

TK
7865
A79
H68
2006

SEQUENTIAL MONTE CARLO METHODS FOR MULTI-SENSOR TRACKING WITH APPLICATIONS TO RADAR SYSTEMS

by

Alon Shalev Housfater
B.Sc., University of Maryland at College Park, 2004

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2006

© Alon Shalev Housfater, 2006

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

UMI Number: EC53498

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform EC53498
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Sequential Monte Carlo Methods for Multi-Sensor Tracking with Applications To Radar Systems

Alon Shalev Housfater

M.A.Sc., Electrical Engineering, Ryerson University, 2006

The aim of this thesis is to explore specific sequential Monte Carlo (SMC) methods and their application to the unique demands of radar and bearing only tracking systems.

Asynchronous radar networks are of special interest and a novel algorithm, the multiple imputation particle filter (MIPF), is formulated to perform data fusion and estimation using asynchronous observations. Convergence analysis is carried out to show that the algorithm will converge to the optimal filter. Simulations are performed to demonstrate the effectiveness of this filter.

Next, the problem of multi-sensor bearing only tracking is tackled. A particle based tracking algorithm is derived and a new filter initialization scheme is introduced for the specific task of multi-sensor bearing only tracking. Simulated data is used to study the efficiency and performance of the initialization scheme.

Acknowledgments

I wish to express my gratitude to several people who were involved in this thesis. I would like to thank my thesis advisor, Dr. Xiao-Ping Zhang, for his comments and insights. His guidance and critical advice were invaluable to me.

I would like to acknowledge and thank Dr. Yifeng Zhou of Defence R&D Canada who introduced me to the topic of radar tracking and provided valuable guidance and advice.

My many friends at CASPAL and elsewhere were a source of support and comfort throughout my studies. I will always think of them fondly.

I cannot end without thanking my family, this thesis would not have been possible without their support, love and encouragement. They were always there for me and it is to them I dedicate this work.

Toronto, August 2006
Alon Shalev

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation and Contributions | 3 |
| 1.2 | Organization | 4 |
| 2 | Target Tracking - Theoretical Aspects | 5 |
| 2.1 | Mathematical Preliminaries | 5 |
| 2.1.1 | Notation | 5 |
| 2.1.2 | Logic and Set Theory | 6 |
| 2.1.3 | Weak Convergence on Metric Spaces | 6 |
| 2.1.4 | Probability Theory | 7 |
| 2.2 | The Nonlinear Dynamic Estimation Problem | 8 |
| 2.3 | Linear Gaussian State Space Models: Kalman Filter | 10 |
| 2.3.1 | Extended Kalman Filter | 11 |
| 2.4 | Sequential Monte Carlo Methods | 12 |
| 2.5 | Convergence of Sequential Monte Carlo Methods | 13 |
| 2.5.1 | Nonlinear Estimation Formulation in Probability Space | 14 |
| 2.5.2 | Particle Filter Formulation in Probability Space | 15 |
| 2.5.3 | Almost Sure Convergence | 16 |
| 3 | Target Tracking for Radar Systems | 21 |
| 3.1 | Problem Formulation | 21 |
| 3.2 | Classical Algorithms | 23 |
| 3.2.1 | Extended Kalman Filter | 23 |
| 3.2.2 | Range Parameterized Extended Kalman Filter | 25 |
| 4 | Multiple Sensor Tracking with Non-Response | 27 |
| 4.1 | Problem Formulation | 27 |
| 4.2 | Multiple Imputations | 28 |
| 4.3 | Multiple Imputation Particle Filter | 30 |
| 4.3.1 | Approximation of the Imputing Function | 30 |
| 4.3.2 | Multiple Imputations Particle Filter Algorithm | 31 |
| 4.4 | Iterative Multiple Imputation Particle Filter | 34 |

| | | |
|----------|--|-----------|
| 4.5 | Convergence Analysis | 34 |
| 4.5.1 | Filter Formulation in Probability Space | 34 |
| 4.5.2 | Almost Sure Convergence | 37 |
| 4.6 | Performance Analysis | 40 |
| 4.6.1 | Radar Network Data Fusion | 40 |
| 4.6.2 | Simulation Results | 41 |
| 5 | Sequential Monte Carlo Methods for Multi-Site Bearing Only Tracking | 45 |
| 5.1 | Problem Formulation | 45 |
| 5.2 | Least Squares Initialization | 46 |
| 5.3 | Bearing only Target tracking with Monte Carlo Algorithms | 48 |
| 5.3.1 | Particle Filter for Bearing Only Tracking | 49 |
| 5.4 | Performance Analysis | 50 |
| 5.4.1 | Simulation Results | 50 |
| 6 | Concluding Remarks | 54 |
| 6.1 | Conclusions | 54 |
| 6.2 | Future Research | 55 |
| | Bibliography | 56 |

List of Figures

| | | |
|-----|--|----|
| 4.1 | Multiple Imputation - Imputation Diagram. | 30 |
| 4.2 | Multiple Imputation Particle Filter - Diagram. | 33 |
| 4.3 | Performance of MIPF and EM for Two Radars - Missing Data Ratio of %50 . | 43 |
| 4.4 | Performance of MIPF and EM for Three Radars - Missing Data Ratio of %25 | 44 |
| 5.1 | Performance of the EKF and Particle filter in Bearing Only Tracking. | 52 |
| 5.2 | Average performance of the EKF and Particle filter in Bearing Only Tracking. | 53 |

Chapter 1

Introduction

Radar systems are complex entities which are composed of several different subsystems such as target identification, track registration [1] and others [2]. One key subsystem is the *tracking subsystem*, this subsystem utilizes a tracking or *filtering* algorithm to estimate the position of the target from noisy radar measurements. The radar traditionally operates in a polar coordinate system due to the mechanics of the radar sensor [2] while the target motion is best modeled in cartesian coordinates. This fact makes the radar tracking problem a nonlinear one due to the coordinate system transition between measurements and state. Nonlinear filtering problems are notoriously difficult to solve due the analytically intractable mathematics. Several approximation schemes have been proposed to resolve this problem such as the extended Kalman filter (EKF) [3], Gaussian sums and numerical integration methods [4]. These approximation methods are often unreliable, prone to filter divergence and their rate of convergence decreases as the dimension of the problem increases. A relatively new approach to nonlinear estimation is the *Sequential Monte Carlo* (SMC) method. SMC methods, also known as *particle filters* (PF), are a set of powerful stochastic algorithms able to compute the optimal filter for a wide range of nonlinear systems. First suggested in [5] and [6], the particle filter uses a randomized or Monte Carlo sampling to approximately obtain the optimal state estimator. There exists significant literature on particle filters since they were first suggested, their performance was further enhanced by the development of the auxiliary particle filter [7] and Rao-Blackwellised

Particle Filter, refer to [8], [9], [10] for a detailed discussion of these topics.

Another problem of fundamental importance is that of *missing data*. Handling missing data has always been a part of statistical analysis, which frequently occurs in various surveys and experiments [11], [12], [13]. Significantly, missing data behavior also appears in the context of data fusion as asynchronous sensors. This can occur in a radar network where the radars might be many miles apart, thus making it difficult to synchronize their observations [1]. There are several standard approaches in dealing with missing data; one is to employ linear prediction techniques to align the data to the instance of missing data, others include formulating a Kalman filter with a time-varying transition matrix that accounts for the missing observations [1]. Also, one can apply the Expectation-Maximization (EM) algorithm to resolve the missing data [12]. These algorithms either assume a linear system or ignore the often nonlinear state dynamics. Thus, available methods are either not applicable to nonlinear systems or fail to incorporate the knowledge of the state dynamics into the algorithm. Conversely, the particle filter does not account for the missing data behavior when performing data fusion and estimation. Therefore, one needs to confront the problem of performing fusion on a nonlinear system in the presence of missing data.

Many radar systems can also operate in a passive mode, while in this mode the radar system measures only the bearing of the target. This type of tracking system is also known as an *Electronic Warfare Device*. Such systems are useful since while standard radar tracking can be detected by sensing the electromagnetic radiation emitted by the radar; the passive radar uses reflections of ambient radio signals to observe the target, thus the target is unaware that it is being tracked. This type of tracking environment is known as *bearing only* target tracking or target motion analysis (TMA). Similar to radar networks, bearing only tracking can be performed by multiple observation stations to increase the reliability and performance of the estimation. As in standard radar tracking, the bearing only tracking is formulated as a nonlinear state space estimation problem. However, unlike radar tracking, bearing only estimation introduces additional problems such as

observability [14], initialization and tracker instability [15]. The problems of initialization and tracker stability are intertwined; sufficiently incorrect filter initialization may cause tracker divergence after a few time iterations. The standard solution to the bearing only initialization problem is to apply the *range parameterized extended Kalman filter* (RPEKF) [16], [17]. The RPEKF consists of a bank of EKFs; thus, there is always a risk of tracker divergence since the EKF is a linearized sub-optimal filter. Therefore, it is of great interest to formulate an initialization procedure that is equivalent to the RPEKF, but applied to the particle filter which has no divergence problems due to linearization.

1.1 Motivation and Contributions

Joint fusion and estimation for nonlinear multi-sensor networks are common tasks in many applications. These tasks can be performed by applying SMC methods, which are a set of powerful methods for nonlinear estimation. However, in many multi-sensor networks, sensors are not synchronized. Asynchronous sensors cause some observations to be missing at any given time instance. SMC methods do not directly account for such system behavior; thus, it is of interest to resolve this dichotomy so one can apply SMC methods in the setting of asynchronous multi-sensor estimation. With these considerations in mind, the *multiple imputation particle filter* (MIPF) is introduced. The MIPF algorithm allows efficient multi-sensor tracking with incomplete observations. The algorithm operates as a bank of particle filters where the multiple imputations (MI) method is used to generate multiple complete observation, each complete observation set is fed to a particle filter. The output of the particle filters are then linearly combined. The problem domain of asynchronous radar networks is explored as an application for the MIPF.

Another commonly encountered estimation problem is bearing only tracking. Many applications use a network of bearing only sensors for tracking uses. However, this problem cannot be solved using a straight forward application of SMC methods due to the inherent instability of the bearing only measurement model. This instability manifests itself as tracker divergence when the filter is initialized with values that are too "differ-

ent" from the true initial state. Thus, a new initialization scheme for the particle filter is suggested, this scheme is applicable for a tracker operating in a multi-sensor bearing only environment. It utilizes several bearing measurements to obtain multiple position estimates using a least squares approach. These estimates are in turn used to construct an initial state distribution which can be used to initialize the particle filter.

1.2 Organization

The organization of this thesis is as follows: Chapter 2 presents a general mathematical development of relevant nonlinear estimation theory. In chapter 3, the discussion focuses on the application domain of radar and bearing only tracking from a nonlinear system perspective. Chapter 4 introduces a new filtering algorithm, the MIPF. Chapter 5 presents a new initialization scheme for multi-sensor bearing only particle filter. The thesis concludes with final remarks and future directions of research.

Chapter 2

Target Tracking - Theoretical Aspects

Understanding sequential Monte Carlo methods require an array of mathematical tools; this section reviews the basic concepts involved in formulating and solving nonlinear estimation problems. First, a brief review of necessary mathematical theory and notation is presented. Next, these mathematical tools are used to formulate the general nonlinear estimation problem and its special case of Gaussian linear state space model. Solutions to the estimation problems such as the particle filter, Kalman filter and EKF are discussed. Finally, the particle filter is analyzed in an abstract setting in order to explore its convergence. The material in this chapter was extracted from various mathematics and signal processing sources [8], [18], [19], [9], [20], [21].

2.1 Mathematical Preliminaries

2.1.1 Notation

Let \mathbb{R} denote the extended real number system, the symbol \mathbb{R}^+ denotes the set of all real positive numbers; $\mathbb{R}^+ = \{r \in \mathbb{R} : r > 0\}$. Similarly, $\mathbb{N} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ denotes the set of integers while \mathbb{N}^+ denotes the set of non-negative integers $\mathbb{N}^+ = \{0, 1, 2, \dots\}$. Bold faced letters such as \mathbf{X} denote vector-valued variables. A variable's dimension is specified in the discussion. A sequence of values v_1, v_2, \dots, v_N is written in shorthand as $\{v_i\}_{i=1}^N$ where i indexes the sequence.

2.1.2 · Logic and Set Theory

Let X be a set, then $x \in X$ indicates that X contains the element x . The symbol \forall denotes 'for all', $\forall x \in X$ means for all members of set X . Let Y be another set such that every member of Y is also in X , then write $Y \subseteq X$, i.e. Y is a subset of X . Consider the two sets X and Y , their union $X \cup Y$ is the set such that any member $x \in X$ and $y \in Y$ is contained in $X \cup Y$. The intersection of the two sets X and Y is defined as the set of members of X and Y where each member is both in X and Y .

2.1.3 Weak Convergence on Metric Spaces

Topological Spaces

A topological space is defined as the set X with a system of subsets τ that contain the null set, the set X itself, the union of every one of its subsystems and the intersection of every one of its finite subsystems. The sets in τ are called the open sets of the topological space (X, τ) . A neighborhood of the point $x \in X$ is the set containing an open set which contains x . The member $x \in X$ is a *limit point* of a subset $M \subseteq X$ if every neighborhood of x contains at least one point $m \in M$ different from x . Concluding this brief discussion of topological spaces, one notes that the critical notion of a topological space allows us to discuss the concept of limit and thus of convergence [20].

Metric Spaces

Consider a set X where for any pair $p, q \in X$, there is a function $d : X \times X \rightarrow \mathbb{R}$ such that the following conditions hold,

1. $d(p, q) \geq 0$ and $d(p, q) = 0$ iff $p = q$.
2. $d(p, q) = d(q, p)$.
3. $d(p, q) \leq d(p, l) + d(q, l)$ where $l \in X$.

One can think of d as a distance measure between a pair of points $p, q \in X$. Call the set (E, d) a *metric space* [20], where d is the *metric function* and E is the ambient space.

Note that a metric space is also a topological space since a metric function d induces a topology in the following way: Let $x_0 \in X, r \in \mathbb{R}^+$ and associate with those two members a set $S(x_0; r) = \{x \in X; d(x, x_0) < r\}$, this set is the open sphere with center x_0 and radius r . It can be shown [20] that the collection of sets $S = \{S(x; r) : x_0 \in X, r \in \mathbb{R}^+\}$ forms a system of open sets and thus (E, S) is a topological space.

Continuous Functions

The symbol $f : X \rightarrow Y$ denotes a function whose domain is the set X and range is the set Y , for every $x \in X$, assign a corresponding element $f(x) \in Y$. Consider two functions, $f : X \rightarrow Y$ and $g : Y \rightarrow Z$, then the composite mapping $g \circ f$ is defined as $(g \circ f)(x) = g(f(x))$. Let $(X, d_x), (Y, d_y)$ be two metric spaces and define the function $f : X \rightarrow Y$. The function f is *continuous* at point $c \in X$ if for every positive real number $\varepsilon \in \mathbb{R}^+$, there exists a positive real number $\delta \in \mathbb{R}^+$ such that for all $x \in X$ satisfying $d_x(x, c) < \delta$ will also satisfy $d_y(f(x), f(c)) < \varepsilon$. The function f is said to be continuous if it is continuous at every point in its domain.

Weak Convergence

A specialized kind of convergence known as *weak convergence* [21] is commonly defined on metric spaces. Consider a metric space (E, d) and a sequence of points in this space $\{e_n \in X\}_{n=1}^{\infty}$, then the sequence e_n weakly converges to the point e , $\lim_{n \rightarrow \infty} e_n = e$ if and only if $\lim_{n \rightarrow \infty} d(e_n, e) = d(e, \varepsilon)$ for all $\varepsilon \in E$.

2.1.4 Probability Theory

A probability space [19] is written as the set triplet $(\Omega, \mathcal{F}, \mathbb{P})$ where Ω is the sample space, \mathcal{F} is a σ -algebra [21] containing all possible events and \mathbb{P} is the probability measure, $\mathbb{P} : \mathcal{F} \rightarrow \mathbb{R}$. The σ -algebra \mathcal{F} is typically taken to be the *Borel* σ -algebra, i.e. the σ -algebra generated by the open sets, of the sample space Ω when possible and is written $\mathcal{B}(\Omega)$. The

density function of X , when it exists, is defined as a function satisfying the relation

$$\mathbb{P}(B) = \int_B f_X(x) dx \text{ for any } B \in \mathcal{F},$$

where B is an arbitrary Borel set in \mathcal{F} . A random variable is a function X with the domain Ω and whose range is contained in \mathbb{R} , then write $X : \Omega \rightarrow \mathbb{R}$. The *expectation* of the random variable X is defined as

$$\mathbb{E}(X) = \int_{\Omega} X(\omega) \mathbb{P}(d\omega).$$

X is \mathbb{P} -integrable, i.e. with respect to the probability measure \mathbb{P} , if and only if the integral above exists and is finite. Consider the collection of random variables $\{X_j\}_{j=1}^N$ defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, they are said to be *independent* if and only if for any collection of Borel sets $\{B_j \in \mathcal{F}\}_{j=1}^N$

$$\mathbb{P}\left(\bigcap_{j=1}^n X_j \in B_j\right) = \prod_{j=1}^N \mathbb{P}(X_j \in B_j).$$

A probability measure of special interest in this work is the *Gaussian* distribution. Let the event space be the n -dimensional Euclidean space, $\Omega = \mathbb{R}^n$ with the σ -algebra, $\mathcal{F} = \mathcal{B}(\mathbb{R}^n)$ then the Gaussian distribution is defined

$$\mathbb{P}(X \in B) = \int_B \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2} (x - \mathbf{m})^T \Sigma^{-1} (\mathbf{m})\right\} dx \text{ for any Borel set } B \in \mathcal{F},$$

where Σ is a symmetric, positive definite *covariance matrix* and \mathbf{m} is the mean vector. It is standard shorthand notation to write this Gaussian distribution as $\mathcal{N}(\mathbf{m}, \Sigma)$.

2.2 The Nonlinear Dynamic Estimation Problem

The first component of any estimation problem is the *state equation*, this equation relates the system's current state to its past state using a stochastic difference equation. Consider an n_x -dimensional discrete-time dynamic system whose time evolution can be described by a vector equation of the form

$$\mathbf{X}_n = f_n(\mathbf{X}_{n-1}, \mathbf{V}_n), \tag{2.1}$$

where \mathbf{X}_n is the n_x -dimensional state vector, f is the nonlinear system transition function and \mathbf{V}_n is the state noise term. Note that by only considering the past state \mathbf{X}_{n-1} , Equation 2.1 makes the implicit assumption that the state sequence $\{\mathbf{X}_n; n \in \mathbb{N}^+\}$ is a Markov process. Also, let the process's initial state be described by an initial probability distribution $\mu(\mathbf{X}_n)$. Assume there exists a *state transition probability density*, $p(\mathbf{X}_n|\mathbf{X}_{n-1})$, which is the conditional density of the probability measure that describes how likely is for the current state value to be \mathbf{X}_n given the past state is \mathbf{X}_{n-1} . A sensor takes noisy observations of the dynamic system (2.1) described above, these measurements are related to the dynamic system's state by the *measurement equation* of the form,

$$\mathbf{Y}_n = g_n(\mathbf{X}_n, \mathbf{W}_n), \quad (2.2)$$

where \mathbf{Y}_n is the n_y -dimensional observation vector, g_n is the nonlinear measurement function and \mathbf{W}_n is the observation noise component. Note that in Equation 2.2 the observation sequence $\{\mathbf{Y}_n; n \in \mathbb{N}^+\}$ is conditionally independent given \mathbf{X}_n . Assume there exists a probability density $p(\mathbf{Y}_n|\mathbf{X}_n)$, which is the density of the probability measure that describes how likely is the measurement \mathbf{Y}_n given that the current state is \mathbf{X}_n . Our aim is to estimate recursively in time the *filtering distribution* $p(\mathbf{X}_n|\mathbf{Y}_n)$ so that one may take various expectations of the form

$$\mathbb{E}(h(\mathbf{X}_n|\mathbf{Y}_n)) = \int h(\mathbf{X}_n) p(\mathbf{X}_n|\mathbf{Y}_n) d\mathbf{X}_n, \quad (2.3)$$

for any $p(\mathbf{X}_n|\mathbf{Y}_n)$ -integrable $h: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$. Common choices for the function h are the minimum mean squared error (MMSE) estimator $h(\mathbf{X}_n) = \mathbf{X}_n$ and the maximum a posteriori (MAP) estimator

$$h(\mathbf{X}_n) = \begin{cases} (1/p(\mathbf{X}_n|\mathbf{Y}_n)) \delta(\mathbf{X}_n) & \text{where } p(\mathbf{X}_n|\mathbf{Y}_n) \text{ is maximized} \\ 0 & \text{for all other } \mathbf{X}_n \end{cases}$$

where δ is the dirac delta function. The general solution for this estimation problem can be written as the following recursive formula,

$$p(\mathbf{X}_n|\mathbf{Y}_n) = p(\mathbf{X}_{n-1}|\mathbf{Y}_{n-1}) \frac{p(\mathbf{Y}_n|\mathbf{X}_n) p(\mathbf{X}_n|\mathbf{X}_{n-1})}{p(\mathbf{Y}_n|\mathbf{Y}_{0:n-1})}. \quad (2.4)$$

However, the denominator of equation 2.4 above cannot usually be evaluated analytically. As a result, one must resort to approximation and numerical integration methods as will be discussed later on in this chapter.

2.3 Linear Gaussian State Space Models: Kalman Filter

Assume that the state transition function f and measurement function g , defined in section 2.2, are linear functions while the noise terms in the state space model are additive Gaussian random processes. Thus, one can rewrite equations 2.1-2.2 as follows

$$\begin{aligned}\mathbf{X}_n &= F_n \mathbf{X}_{n-1} + \mathbf{V}_n \\ \mathbf{Y}_n &= G_n \mathbf{X}_n + \mathbf{W}_n,\end{aligned}$$

where F_n is a $n_x \times n_x$ matrix, G_n is an $n_y \times n_x$ matrix and $\mathbf{V}_n \sim \mathcal{N}(0, R_n)$, $\mathbf{W}_n \sim \mathcal{N}(0, Q_n)$ are the independent identically distributed (*i.i.d.*) Gaussian state noise and observation noise respectively. Given the assumptions of the linear Gaussian state space model, it can be shown [22] that the filtering probability density function $p(\mathbf{X}_n | \mathbf{Y}_{0:n})$ is also Gaussian. Since the filtering density is Gaussian, it is characterized by its first and second moments [19], which can be written

$$\begin{aligned}\hat{\mathbf{X}}_{n|n} &= \mathbb{E}(\mathbf{X}_n | \mathbf{Y}_{0:n}) \\ \mathbf{P}_{n|n} &= \mathbb{E}\left(\left(\mathbf{X}_n - \hat{\mathbf{X}}_{n|n}\right)\left(\mathbf{X}_n - \hat{\mathbf{X}}_{n|n}\right)^T | \mathbf{Y}_{0:n}\right),\end{aligned}$$

where $\hat{\mathbf{X}}_{n|n}$ is the estimate of the state at time n and $\mathbf{P}_{n|n}$ is its associated covariance matrix. In order to obtain these estimates one can apply the well known Kalman filter [3] in two stages; first predict the state using information from time instance $n - 1$,

$$\begin{aligned}\hat{\mathbf{X}}_{n|n-1} &= F_n \hat{\mathbf{X}}_{n-1|n-1} \\ \mathbf{P}_{n|n-1} &= F_n \mathbf{P}_{n-1|n-1} F_n^T + Q_n,\end{aligned}$$

then update these prediction estimates with the current measurement at time n as follows

$$\begin{aligned}\mathbf{K}_n &= \mathbf{P}_{n|n-1}G_n^T G_n \mathbf{P}_{n|n-1}G_n^T + R_n \\ \widehat{\mathbf{X}}_{n|n} &= \widehat{\mathbf{X}}_{n|n-1} + \mathbf{K}_n \left(\mathbf{Y}_n - G_n \widehat{\mathbf{X}}_{n|n-1} \right) \\ \mathbf{P}_{n|n} &= (\mathbf{I} - \mathbf{K}_n G_n) \mathbf{P}_{n|n-1}.\end{aligned}$$

Note that under the linear Gaussian assumptions above, the Kalman filter is the optimal MMSE state estimator [22].

2.3.1 Extended Kalman Filter

While the Kalman filter is optimal for linear target dynamics, it is not applicable for non-linear system dynamics. One approach to filter nonlinear systems is to linearize the state space model and then apply the standard Kalman filter on the linearized model. This approach is known as the extended Kalman filter (EKF) [2]. Consider a state space model of the following form,

$$\mathbf{X}_n = f_n(\mathbf{X}_{n-1}) + \mathbf{V}_n \quad (2.5)$$

$$\mathbf{Y}_n = g_n(\mathbf{X}_n) + \mathbf{W}_n, \quad (2.6)$$

where all symbols are as defined in previous sections. Define the jacobians of the state and observation transition functions,

$$\begin{aligned}\mathbf{J}_n^f &\triangleq \left. \frac{\partial f_n}{\partial \mathbf{x}} \right|_{\mathbf{x}=\widehat{\mathbf{X}}_{n|n-1}} \\ \mathbf{J}_n^g &\triangleq \left. \frac{\partial g_n}{\partial \mathbf{x}} \right|_{\mathbf{x}=\widehat{\mathbf{X}}_{n|n-1}},\end{aligned}$$

then by taking a vector Taylor series expansion of model 2.5-2.6 one obtains the linearized state space model

$$\mathbf{X}_n = f_n(\widehat{\mathbf{X}}_{n-1|n-1}) + \mathbf{J}_n^f \left(\mathbf{X}_{n-1} - \widehat{\mathbf{X}}_{n-1|n-1} \right) + \mathbf{V}_n \quad (2.7)$$

$$\mathbf{Y}_n = g_n(\widehat{\mathbf{X}}_{n|n-1}) + \mathbf{J}_n^g \left(\mathbf{X}_n - \widehat{\mathbf{X}}_{n|n-1} \right). \quad (2.8)$$

Clearly, equations 2.7 - 2.8 are a linear Gaussian state space model and therefore the Kalman filter can be applied.

2.4 Sequential Monte Carlo Methods

Due to the problems associated with the EKF, an alternative approach was suggested in the form of the particle filter. A particle filter adopts a Monte Carlo sampling strategy in order to approximate the filtering distribution $p(\mathbf{X}_{0:n}|\mathbf{Y}_n)$. It is approximated by a set of N particles, or possible values of the state, $\{x_n^i\}_{i=1}^N$ with associated weighting coefficients $\{w_n^i\}_{i=1}^N$ where i indexes the particles at time n . The filtering distribution's particle approximation can be written as the sum,

$$p(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n}) \simeq \sum_{i=1}^N w_n^i \delta(\mathbf{X}_{0:n} - x_{0:n}^i). \quad (2.9)$$

Thus one can take expectations as in Equation 2.3,

$$\mathbb{E}(h(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n})) = \int h(\mathbf{X}_{0:n}) p(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n}) d\mathbf{X}_{0:n} \simeq \sum_{i=1}^N w_n^i h(x_{0:n}^i).$$

The particles are sampled from an *Importance Function* which is denoted $\pi(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n})$. This Monte Carlo technique of *Importance Sampling* works by sampling N states from the importance function $\pi(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n})$,

$$x_{0:n}^i \sim \pi(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n}), \quad (2.10)$$

and evaluate the i -th particle's weighting coefficient using the formula

$$\tilde{w}_n^i = \frac{p(\mathbf{Y}_{0:n}|\mathbf{X}_{0:n}) p(\mathbf{X}_{0:n})}{\pi(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n})}. \quad (2.11)$$

Note that the weighting coefficient \tilde{w}_n^i is un-normalized. It is normalized using the formula $w_n^i = \tilde{w}_n^i / \sum_{j=1}^N \tilde{w}_n^j$ in order to make the approximate density a probability measure. The procedure is discussed so far generates an approximation of the distribution $p(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n})$, however this is not a recursive procedure. In other words, one must sample the entire *state trajectory* $x_{0:n}^i$ to obtain the approximation given in equation 2.9. In order to turn this procedure into a sequential Monte Carlo filter, assume the importance function has the following decomposition

$$\pi(\mathbf{X}_{0:n}|\mathbf{Y}_{0:n}) = \pi(\mathbf{X}_0|\mathbf{Y}_0) \prod_{k=1}^n \pi(\mathbf{X}_k|\mathbf{X}_{0:k-1}, \mathbf{Y}_{0:k}). \quad (2.12)$$

Using this assumption, one can modify the Monte Carlo approximation 2.10-2.11 into a recursive procedure where at time n it operates as follows,

$$\begin{aligned} x_n^i &\sim \pi(\mathbf{X}_n | x_{0:n-1}^i, \mathbf{Y}_{0:n}) \\ \tilde{w}_n^i &= \frac{p(\mathbf{Y}_n | x_n^i) p(x_n^i | x_{n-1}^i)}{\pi(x_n^i | x_{0:n-1}^i, \mathbf{Y}_{0:n})}, \end{aligned}$$

and then normalize the weights $w_n^i = \tilde{w}_n^i / \sum_{j=1}^N \tilde{w}_n^j$ to obtain the particle set $\{x_{0:n}^i, w_n^i\}_{i=1}^N$. The importance function itself may be any function that obeys the condition 2.12, however there are several standard choices. One is the *prior importance function* where one uses the state transition density as the importance function, $\pi(\mathbf{X}_{0:n} | \mathbf{Y}_{0:n}) = p(\mathbf{X}_n | x_{n-1}^i)$. The *optimal importance function* is the function which minimizes the variance of the importance weights, it can be shown [8] that this function is $p(\mathbf{X}_n | x_{n-1}^i, \mathbf{Y}_n)$. The particle filter can be written in algorithm notation as follows.

The Particle Filter Algorithm

- Initialize the particles $x_0^i \sim \mu(\mathbf{X}_0)$ and set $w_0^i = 1/N$ for $i = 1, \dots, N$.
 - For $n = 1, 2, \dots$
 - For $i = 1, \dots, N$ Sample the importance function $x_n^i \sim \pi(\mathbf{X}_n | \mathbf{Y}_{0:n})$
 - Evaluate the importance weights: $\tilde{w}_n^i = \frac{p(\mathbf{Y}_n | x_n^i) p(x_n^i | x_{n-1}^i)}{\pi(x_n^i | x_{0:n-1}^i, \mathbf{Y}_{0:n})}$ and normalize $w_n^i = \tilde{w}_n^i / \sum_{j=1}^N \tilde{w}_n^j$
 - Sample the index $d(i)$ distributed according to discrete distribution such that $\mathbb{P}(d(i) = l) = w_n^l$ for $l = 1, \dots, N$.
 - Set $x_n^i = x_n^{d(i)}$ and $w_n^i = 1/N$ for $i = 1, \dots, N$.
-

2.5 Convergence of Sequential Monte Carlo Methods

In the previous section, a particle filtering scheme was discussed, this method uses a large number of particles to approximate the filtering distribution. However, the issue

of convergence remains; more explicitly, let N be the number of particles, then one is interested in exploring whether the distribution approximation converges as N increases; these questions of convergence are discussed in this section. The particle filter is reformulated to make it more amenable to convergence analysis. Next, its almost sure weak convergence is analyzed and proved.

2.5.1 Nonlinear Estimation Formulation in Probability Space

Following the development in [18] let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space where $\mathcal{F} = \mathcal{B}(\mathbb{R}^{n_x})$ is the Borel set of \mathbb{R}^{n_x} . On this probability triple define a vector-valued stochastic process $X = \{X_n, n \in \mathbb{N}^+\}$ where n_x is the dimension of the state space of X . The process X is Markov with initial distribution μ and probability transition kernel $K(x_n|x_{n-1})$

$$\mathbb{P}(X_n \in A | X_{n-1} = x_{n-1}) = \int_A K(x_n|x_{n-1}) dx, A \in \mathcal{B}(\mathbb{R}^{n_x}).$$

The process X can be viewed as hidden state process to be estimated. Next, define a stochastic process $Y = \{Y_n, n \in \mathbb{N}^+\}$ where n_y is the dimension of the state space of Y . The process Y is conditionally independent of X

$$\mathbb{P}(Y_n \in B | X_n = x_n) = \int_B g(y_n|x_n) d\omega_n, B \in \mathcal{B}(\mathbb{R}^{n_w \times k}).$$

The stochastic process Y can be considered a noisy observation of the hidden Markov process X . Define the family of probability distributions,

$$p(x_{k:l}|y_{1:m}) = \mathbb{P}(X_k \in dx_k, \dots, X_l \in dx_l | Y_1, \dots, Y_m).$$

Bayes' theorem lets use write the joint distribution of the state at time n ,

$$p(x_{0:t}) = \mu(x_0) \prod_{k=1}^n K(x_k|x_{k-1}) g(y_k|x_k),$$

and the recursive equations

$$p(x_{0:n}|y_{1:n-1}) = p(x_{0:n-1}|y_{1:n-1}) K(x_n|x_{n-1}) \tag{2.13}$$

$$p(x_{0:n}|y_{1:n}) = \left[\int_{\mathbb{R}^{n_x}} g(y_n|x_n) p(x_{0:n}|y_{1:n-1}) \right]^{-1} g(y_n|x_n) p(x_{0:n}|y_{1:n-1}). \tag{2.14}$$

As mentioned in section 2.2, one is typically interested in the marginal distribution $p(x_n|y_n)$, using the equations 2.13-2.14 one can write,

$$p(x_n|y_{n-1}) = \int_{\mathbb{R}^{n_x}} p(x_{n-1}|y_{n-1}) K(x_n|x_{n-1}) dx_{n-1} \quad (2.15)$$

$$p(x_n|y_n) = \left[\int_{\mathbb{R}^{n_x}} g(y_n|x_n) p(x_n|y_{n-1}) \right]^{-1} g(y_n|x_n) p(x_n|y_{n-1}). \quad (2.16)$$

It will prove useful to rewrite the equations above in a different way, let φ be a function defined as $\varphi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ and ν a measure, then using the standard notation,

$$(\nu, \varphi) = \int \varphi \nu,$$

one can rewrite 2.15-2.16,

$$(p(x_n|y_{n-1}), \varphi) = (p(x_n|y_n), K\varphi)$$

$$(p(x_n|y_n), \varphi) = (p(x_n|y_{n-1}), g)^{-1} (p(x_n|y_{n-1}), \varphi g).$$

2.5.2 Particle Filter Formulation in Probability Space

As discussed in section 2.4, the particle filtering method recursively approximates the filtering distribution $p(x_n|y_n)$ at time n . The approximation is done by generating a "cloud" of N particles $\{x_n^i\}_{i=1}^N$ that generate an empirical measure $p^N(x_n|y_n)$ as follows

$$p^N(x_n|y_n) = \frac{1}{N} \sum_{i=1}^N \delta(x_n - x_n^i),$$

where δ denotes the dirac delta function. The algorithm is recursive in the sense that the particles at time $n-1$, $\{x_{n-1}^i\}_{i=1}^N$, are used to generate the particles at time n , $\{x_n^i\}_{i=1}^N$ and thus obtain the empirical measure $p^N(x_n|y_n)$. The basic algorithm proceeds as follows; assume that at time n there are N particles distributed approximately according to $p(x_{n-1}|y_{n-1})$. Then one samples $\tilde{x}_n^i \sim K(x_n|x_{n-1}^i)$. These new particles are distributed approximately according to $p(x_n|y_{n-1})$ [3]. The empirical distribution can be written,

$$\tilde{p}^N(x_n|y_{n-1}) = \frac{1}{N} \sum_{i=1}^N \delta(x_n - \tilde{x}_n^i),$$

which is an approximation of $p(x_n|y_{n-1})$. Substituting the empirical measure $\tilde{p}^N(x_n|y_{n-1})$ to Equation 2.16, one obtains the Monte Carlo approximation,

$$\tilde{p}^N(x_n|y_n) = \left[\sum_{i=1}^N g(y_n|\tilde{x}_n^i) \right]^{-1} g(y_n|\tilde{x}_n^i) \delta(x_n - \tilde{x}_n^i),$$

the empirical distribution $\tilde{p}^N(x_n|y_n)$ approximates the desired filtering distribution $p(x_n|y_n)$,

$$\tilde{p}^N(x_n|y_n) = \sum_{i=1}^N w_n^i \delta(x_n - x_n^i), \quad \sum_{i=1}^N w_n^i = 1,$$

where $w_n^i \propto g(y_n|\tilde{x}_n^i)$ are the *importance weights*. The particle filter performs a *resampling* procedure to obtain an un-weighted empirical distribution,

$$p^N(x_n|y_n) = \frac{1}{N} \sum_{i=1}^N \delta(x_n - x_n^i),$$

by removing particles with low weights and propagating particles with high weight. The resampling is typically done by sampling the weighted empirical distribution, $x_n^i \sim \tilde{p}^N(x_n|y_n)$. It turns out that the resampling stage is a critical algorithmic step that stabilizes the algorithm so that increasing number of particles are not necessary as time progresses [8].

2.5.3 Almost Sure Convergence

Preliminaries

Consider a metric space (E, d) and let $\{a_n\}_{n=1}^\infty, \{b_n\}_{n=1}^\infty$ be two sequences of continuous functions indexed by $n \in \mathbb{N}^+$, write these sequence of functions $a_n, b_n : E \rightarrow E$. Moreover, let there be two other sequences of functions $k_n, k_{1:n}$ as follows

$$\begin{aligned} k_n &= a_n \circ b_n \\ k_{1:n} &= k_n \circ k_{n-1} \circ \dots \circ k_1, \end{aligned}$$

the functions k_n and $k_{1:n}$ are perturbed by the function $c^N : E \rightarrow E$,

$$\begin{aligned} k_n^N &= c^N \circ a_n \circ c^N \circ b_n \\ k_{1:n}^N &= k_n^N \circ k_{n-1}^N \circ \dots \circ k_1^N. \end{aligned}$$

Assume that as N increases, the perturbation becomes smaller, i.e. $\lim_{N \rightarrow \infty} c^N = I$ where I is the identity function on E . One wishes to know whether given that c^N will converge in a predefined manner, will $k_{1:t}^N$ converge to $k_{1:t}$. It turns out that in order for $k_{1:t}^N$ to converge, the perturbation function c^N must converge in a uniform manner [18], [20]. Thus, c^N needs to satisfy the following condition

$$\lim_{N \rightarrow \infty} e_N = e \Rightarrow \lim_{N \rightarrow \infty} c^N(e_N) = e, \quad (2.17)$$

for all sequences $e_N, e \in E$. Assuming condition 2.17, one can prove [18] the following lemma

Lemma 1 *Let $a_n, b_n, k_n, k_{1:n}$ and c^N be as defined above, then if c^N satisfies condition 2.17, the following is true,*

$$\lim_{N \rightarrow \infty} k_n^N = k_n \text{ and } \lim_{N \rightarrow \infty} k_{1:n}^N = k_{1:n}.$$

Convergence of the Particle Filter

We discussed some convergence properties of functions on abstract spaces. We now relate these abstract concepts with the particle filter in order to establish its convergence. Let $E = \mathcal{P}(\mathbb{R}^{n_x})$ be the space of probability measures over the n_x -dimensional Euclidean space \mathbb{R}^{n_x} . We endow the space with the topology of weak convergence. In this topology, the convergence on the space is defined as follows, consider a sequence of probability measures $\{\nu_N \in \mathcal{P}(\mathbb{R}^{n_x}) : N = 1, \dots, \infty\}$, then we say the sequence $\nu_N \in \mathcal{P}(\mathbb{R}^{n_x})$ weakly converges to ν ($\lim_{N \rightarrow \infty} \nu_N = \nu$) if and only if, for any continuous bounded function $\varphi \in C_b(\mathbb{R}^{n_x})$

$$\lim_{N \rightarrow \infty} (\nu_N, \varphi) = (\nu, \varphi),$$

where $C_b(\mathbb{R}^{n_x})$ is the space of all continuous bounded functions on \mathbb{R}^{n_x} . In other words,

$$\lim_{N \rightarrow \infty} \nu_N = \nu \text{ weakly} \Leftrightarrow \lim_{N \rightarrow \infty} (\nu_N, \varphi) = (\nu, \varphi) \forall \varphi \in C_b(\mathbb{R}^{n_x}).$$

For the sake of completeness, note that it is a well known result that one can choose a countable set $\mathcal{A} = \{\varphi_i\}_{i>0} \in C_b(\mathbb{R}^{n_x})$ so that it completely determines convergence. Thus,

one can write

$$\lim_{N \rightarrow \infty} \nu_N = \nu \text{ weakly} \Leftrightarrow \lim_{N \rightarrow \infty} (\nu_N, \varphi_i) = (\nu, \varphi) \forall \varphi_i \in \mathcal{A}.$$

Using the countable set \mathcal{A} , one can define the metric d associated with this space and topology,

$$d(a, b) = \sum_{i=1}^{\infty} \frac{|(a, \varphi_i) - (b, \varphi_i)|}{2^i \|\varphi_i\|},$$

where $\|\varphi_i\|$ is defined as the supremum norm of φ_i on $C_b(\mathbb{R}^{n_x})$.

Now that the convergence space is established, let us proceed by identifying the abstract function sequences a_n, b_n with particle filtering operations. Let the sequence of functions $b_n : \mathcal{P}(\mathbb{R}^{n_x}) \rightarrow \mathcal{P}(\mathbb{R}^{n_x})$ be the mappings

$$b_n(\nu) = \int_{\mathbb{R}^{n_x}} K(x_n | x_{n-1}) \nu dx_{n-1}, \quad (2.18)$$

for arbitrary $\nu \in \mathcal{P}(\mathbb{R}^{n_x})$. Thus, the following formula is true

$$p(x_n | y_{n-1}) = b_n(p(x_{n-1} | y_{n-1})),$$

since $p(x_{n-1} | y_{n-1}) \in \mathcal{P}(\mathbb{R}^{n_x})$. Recall from section 2.5.3 the sequence of functions, b_n , is defined as a sequence of continuous functions, thus to ensure that the mapping (2.18) is continuous one assumes that the transition kernel of the signal, $K(x_n | x_{n-1})$, is *Feller* [19]. The property of a kernel being *Feller* is defined as that for any continuous bounded function φ , $K\varphi$ is also continuous, i.e.

$$\varphi \in C_b(\mathbb{R}^{n_x}) \Rightarrow K\varphi \in C_b(\mathbb{R}^{n_x}).$$

Given the assumption of Feller property and the definition 2.18, one can show that the sequence of functions b_n have the following convergence property,

$$\lim_{N \rightarrow \infty} (b_n(\nu_N), \varphi) = (b_n(\nu), \varphi), \forall \varphi \in C_b(\mathbb{R}^{n_x}).$$

Define the sequence of functions a_n be the mappings $a_n : \mathcal{P}(\mathbb{R}^{n_x}) \rightarrow \mathcal{P}(\mathbb{R}^{n_x})$ such that

$$(a_n(\nu), \varphi) = (\nu, g)^{-1}(\nu, \varphi g) \text{ for any } \varphi \in C_b(\mathbb{R}^{n_x}). \quad (2.19)$$

This definition implies

$$p(x_n|y_n) = a_n(p(x_n|y_{n-1})) = a_n \circ b_n(p(x_{n-1}|y_{n-1})).$$

In order to ensure that the mapping a_n is continuous, assume that the function $g(y_n|\cdot)$ is a continuous bounded strictly positive function

$$g(y_n|\cdot) \in C_b(\mathbb{R}^{n_x}), g(y_n|x_n) > 0 \forall x_n \in \mathbb{R}^{n_x}. \quad (2.20)$$

The positivity assumption is necessary so that the term (ν, g) in Definition 2.19 is never zero. If the function $g(y_n|\cdot)$ satisfies condition 2.20 and $\lim_{N \rightarrow \infty} \nu_N = \nu$, it implies

$$\lim_{N \rightarrow \infty} (a_n(\nu_N), \varphi) = (a_n(\nu), \varphi).$$

In the context of particle filtering, c^N is a stochastic perturbation that occurs due to the sampling of the importance function as discussed in section 2.5.2. Define the random perturbation function c^N as follows

$$c^N(\nu) = \frac{1}{N} \sum_{j=1}^N \delta(V_j),$$

where $\{V_j\}_{j=1}^N$ is a collection of random variables with common distribution ν and $N > 0$ is an integer. The following lemma can be shown to be true:

Lemma 2 *If c^N is defined as above, then it satisfies condition 2.17 almost surely.*

Given the definitions for a_n, b_n and c^N , one can see that the following formula are true

$$\begin{aligned} p^N(x_n|y_n) &= c^N \circ a_n \circ c^N \circ b_n(p(x_{n-1}|y_{n-1})) = k_n(p(x_{n-1}|y_{n-1})) \\ p^N(x_n|y_n) &= k_{1:n}^N \circ c^N(\mu) = k_{1:n}^N((\mu^N)), \end{aligned}$$

where $\mu^N = c^N(\mu)$. Knowing this result and using lemmas 2 and 1, the following theorem can be proven

Theorem 1 *Assuming the transition kernel K is Feller and that the likelihood function g is bounded, continuous and strictly positive, then $\lim_{N \rightarrow \infty} p^N(x_n|y_n) = p(x_n|y_n)$ almost surely.*

In conclusion, a result is obtained that indicates the particle filter will convergence with the true optimal filtering distribution with probability 1 as the number of particles increases toward infinity. However, one must note that this is a weak result since there is no guarantee that a *finite* number of particles will convergence or nor does it guarantee that the particle filter will generate an approximation with bounded error. Thus, this convergence results suggests the particle filter is well behaved but more work is necessary to understand the behavior of the true particle filter. It turns out stronger assumptions are needed in order to guarantee the strong convergence of the particle filter (in the sense of bounded error for finite number of particles). Refer to the work at [23] for a complete and in-depth discussion of convergence results for particle filters.

Chapter 3

Target Tracking for Radar Systems

In this chapter, the radar and bearing only tracking are discussed from the nonlinear system perspective. Due to the significant nonlinearities of the radar tracking and bearing only problems, classical approximation techniques must pay special consideration to issues such as observability and tracker initialization. First, the problems of radar and bearing only tracking are formulated. Next, well known strategies such as the EKF and RPEKF are discussed. The material in this chapter was taken chiefly from references [16], [24], [8], [1], [25].

3.1 Problem Formulation

Assume the target motion is well modeled by linear state dynamics. These dynamics are obtained by considering the target's state in cartesian coordinates. A common target motion model is the *near constant velocity* model [1]. For this model, the state transition equation can be written as the matrix equation,

$$\mathbf{X}_n = \Phi_n \mathbf{X}_{n-1} + \Gamma_n \mathbf{V}_n,$$

where

$$\Phi_n = \begin{pmatrix} 1 & \Delta t_n & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t_n \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and

$$\Gamma_n = \begin{pmatrix} \Delta t_n^2/2 & 0 \\ \Delta t_n & 0 \\ 0 & \Delta t_n^2/2 \\ 0 & \Delta t_n \end{pmatrix}.$$

The state vector \mathbf{X}_n is defined as $\mathbf{X}_n \triangleq [x_n, x'_n, y_n, y'_n]^T$ where x_n, y_n are the cartesian coordinates of the target's position and x'_n, y'_n are associated velocity components. The two-dimensional system noise $\mathbf{V}_n \sim \mathcal{N}(\mathbf{0}, \Sigma_v)$ is assumed to be Gaussian *i.i.d.* process. The symbol Δt_n denotes the time difference between two measurements at time n . Note the distinction between the integer $n \in \mathbb{N}^+$ which indexes the state sequence and the "analog" time difference $\Delta t_n \in \mathbb{R}^+$. The timing difference between successive measurements, Δt_n is assumed to be a time varying model parameter. It is clear that the state transition matrix Φ_n uses Newton's laws of motions in a straightforward manner to transition from the past state to the current one. More interesting, note that the matrix Γ_n turns the state process into a non-homogenous Markov process by making the noise variance time varying. The noise transition matrix models the phenomena that as more time passes between successive measurements, the less deterministic and more random the state trajectory becomes.

Assume the radar is located at $(x_{s,n}, y_{s,n})$ where $x_{s,n}, y_{s,n}$ are the cartesian coordinates of the sensor location at time n . The sensors observations \mathbf{Y}_n are traditionally taken in polar coordinates [25], these *sensor coordinates* cause the state model to become nonlinear due to this coordinate change. The observation model is written,

$$\mathbf{Y}_n = \begin{pmatrix} \theta_n \\ r_n \end{pmatrix} = \begin{pmatrix} \arctan(y_n - y_{s,n}) / (x_n - x_{s,n}) \\ ((x_n - x_{s,n})^2 + (y_n - y_{s,n})^2)^{1/2} \end{pmatrix} + \mathbf{W}_n, \quad (3.1)$$

where $\theta_n, r_n \in \mathbb{R}$ are the bearing and range measurements at time n , respectively. The system noise $\mathbf{W}_n \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$ is a Gaussian *i.i.d.* process.

In bearing only tracking, the bearing θ is available to perform tracking with. Therefore, one rewrites the observation model 3.1 as follows,

$$Y_n = \theta_n = \arctan(y_n - y_{s,n}) / (x_n - x_{s,n}) + W_n. \quad (3.2)$$

Where the noise W_n is a Gaussian random variable and Y_n denotes the noisy bearing observation. Note that the bearing only problem is far more difficult than the standard radar tracking problem. Indeed, if the sensor is stationary, (i.e. $y_{s,n} = y_{s,m}, x_{s,n} = x_{s,m}$ for all $n, m \in \mathbb{N}^+$), the state \mathbf{X}_n is not observable [26]. It can be shown that problem is observable if and only if the sensor's motion has at least one more non-zero derivative compared to the target's motion; in the case of near constant velocity, this implies that the sensor must maneuver in an accelerating manner. This problem of observability for a single sensor disappears if multiple sensors observe the target, i.e., the state space system is always observable assuming multiple bearing readings [26]. The model above 3.2 can be easily extended to K multiple sensors by stacking the multiple observations

$$\mathbf{Y}_n = \begin{pmatrix} Y_n^1 \\ Y_n^2 \\ \vdots \\ Y_n^K \end{pmatrix} = \begin{pmatrix} \arctan \left(\frac{y_n - y_{s,n}^1}{x_n - x_{s,n}^1} \right) \\ \arctan \left(\frac{y_n - y_{s,n}^2}{x_n - x_{s,n}^2} \right) \\ \vdots \\ \arctan \left(\frac{y_n - y_{s,n}^K}{x_n - x_{s,n}^K} \right) \end{pmatrix} + \mathbf{W}_n,$$

where Y_n^i denotes observation made by sensor i at time n and $(y_{s,n}^i, x_{s,n}^i)$ denotes the i -th sensor's cartesian coordinate at time n .

3.2 Classical Algorithms

3.2.1 Extended Kalman Filter

Since the tracking problems discussed above are nonlinear, the standard Kalman filter cannot be used directly. Instead, the Kalman filter ([24], [27]) operates on a linearized version of the observation model 3.1, this filtering strategy is known as the EKF ([24],

[27]). Using notation introduced in section 2.3.1, calculate the Jacobians

$$\mathbf{J}_n^f = \Phi_n$$

$$\mathbf{J}_n^g = \begin{pmatrix} \frac{-y_n - y_{s,n}}{(y_n - y_{s,n})^2 + (x_n - x_{s,n})^2} & \left(((x_n - x_{s,n})^2 + (y_n - y_{s,n})^2)^{1/2} \times (x_n - x_{s,n}) \right)^{-1} \\ 0 & 0 \\ \left(x_n - x_{s,n} + \frac{(y_n - y_{s,n})^2}{x_n - x_{s,n}} \right)^{-1} & \left(((x_n - x_{s,n})^2 + (y_n - y_{s,n})^2)^{1/2} \times (y_n - y_{s,n}) \right)^{-1} \\ 0 & 0 \end{pmatrix}^T,$$

and obtain the linearized observation model,

$$\mathbf{X}_n = \Phi_n \mathbf{X}_{n-1} + \Gamma_n \mathbf{V}_n$$

$$\mathbf{Y}_n = \begin{pmatrix} \arctan(\hat{y}_{n|n-1} - y_{s,n}) / (\hat{x}_{n|n-1} - x_{s,n}) \\ \left((\hat{x}_{n|n-1} - x_{s,n})^2 + (\hat{y}_{n|n-1} - y_{s,n})^2 \right)^{1/2} \end{pmatrix} + \mathbf{J}_n^g (\mathbf{X}_n - \hat{\mathbf{X}}_{n|n-1}),$$

where $\hat{\mathbf{X}}_{n|n-1} \triangleq [\hat{x}_{n|n-1}, \hat{x}'_{n|n-1}, \hat{y}_{n|n-1}, \hat{y}'_{n|n-1}]^T$ is the state prediction at time n as discussed in section 2.3.1. Applying the standard Kalman filter, the prediction stage is,

$$\hat{\mathbf{X}}_{n|n-1} = \mathbf{J}_n^f \hat{\mathbf{X}}_{n-1|n-1}$$

$$\mathbf{P}_{n|n-1} = \mathbf{J}_n^f \mathbf{P}_{n-1|n-1} (\mathbf{J}_n^f)^T + \Gamma_n^T \Sigma_v \Gamma_n,$$

paired with the estimation update stage,

$$\mathbf{K}_n = \mathbf{P}_{n|n-1} (\mathbf{J}_n^g)^T \mathbf{J}_n^g \mathbf{P}_{n|n-1} (\mathbf{J}_n^g)^T + R_n$$

$$\hat{\mathbf{X}}_{n|n} = \hat{\mathbf{X}}_{n|n-1} + \mathbf{K}_n (\mathbf{Y}_n - \mathbf{J}_n^g \hat{\mathbf{X}}_{n|n-1})$$

$$\mathbf{P}_{n|n} = (\mathbf{I} - \mathbf{K}_n \mathbf{J}_n^g) \mathbf{P}_{n|n-1}.$$

Similarly, one can derive the EKF for bearing only tracking 3.2,

$$\mathbf{J}_n^f = \Phi_n$$

$$\mathbf{J}_n^g = \begin{pmatrix} \frac{-y_n - y_{s,n}}{(y_n - y_{s,n})^2 + (x_n - x_{s,n})^2} & 0 & \left(x_n - x_{s,n} + \frac{(y_n - y_{s,n})^2}{x_n - x_{s,n}} \right)^{-1} & 0 \end{pmatrix},$$

with a linearized state space model

$$\mathbf{X}_n = \Phi_n \mathbf{X}_{n-1} + \Gamma_n \mathbf{V}_n$$

$$\mathbf{Y}_n = \arctan(\hat{y}_{n|n-1} - y_{s,n}) / (\hat{x}_{n|n-1} - x_{s,n}) + \mathbf{J}_n^g (\mathbf{X}_n - \hat{\mathbf{X}}_{n|n-1}).$$

The EKF for bearing only tracking is similar to the one discussed above where one substitutes the radar problem's measurement Jacobian, J_n^g , with the Jacobian associated with the bearing only measurement model.

3.2.2 Range Parameterized Extended Kalman Filter

A significant issue with the EKF is the problem of divergence due to incorrect initialization. The EKF must be initialized with an initial range estimate, however, this estimate can be highly inaccurate since only the bearing information is available. As a consequence, the EKF can diverge after a few time iterations regardless of the accuracy of the bearing observations [15]. Researchers attempted to resolve this range initialization problem by introducing the RPEKF [16], [17]. This filter is constructed as a filter bank of N independent EKFs, where each EKF is initialized with a different range estimate. After a few time iterations one can apply convergence diagnostics to determine which EKF is non-divergent and eliminate the diverging filters. This is done by first picking a *coefficient of variation* C_R for each EKF which governs the stability of the associated filter. Usually the initial range is known to be between two limits, $0 \leq r_{min} \leq r_{max}$. One can compute a sequence of ranges $\{r_i\}_{i=1}^N$ that progresses from r_{min} to r_{max} and assign a sequence member to each EKF. The standard way of obtaining this range sequence [16] is to use the geometric progression

$$r_i = \frac{r_{min}}{2} (\rho_i + \rho_{i-1}), \rho_i = \left(\frac{r_{max}}{r_{min}} \right)^{1/N},$$

where r_i is the range used to initialize the i -th EKF. Also, define a coefficient of variation C_R for each EKF to obtain the associated standard deviation $\sigma_i = r_i/C_R$. Given an initial bearing measurement θ_0 , initialize the EKF bank as follows

$$x_0^i = \begin{pmatrix} r_i \sin \theta_0 \\ r_i \cos \theta_0 \\ X' \\ Y' \end{pmatrix},$$

the initial covariance matrix $P_{i,0}$ is also initialized using θ_0 and r_i . At time n , the results of each EKF are combined by keeping track of a weighting coefficient $\gamma_{i,n}$ that is updated

recursively

$$\gamma_{i,n} = \gamma_{i,n-1} p(y_n|i),$$

initially, it is assumed that all EKFs are equally valid and set all weighting coefficients $\gamma_{i,0} = 1/N$. The EKF likelihood density function $p(y_n|i)$ is evaluated by assuming a Gaussian distribution

$$p(y_n|i) \propto \frac{1}{\sqrt{\det(S_{i,n})}} \exp -\frac{1}{2} (y_n - f_n(\hat{x}_{i,n|n-1}))^T S_{i,n}^{-1} (y_n - f_n(\hat{x}_{i,n|n-1}))$$

$$S_{i,n} = H_{i,n} P_{i,n|n-1} H_{i,n}^T + R_n,$$

once the weighting coefficients $\gamma_{i,n}$ are obtained, the estimation and covariance matrix are calculated

$$\hat{x}_{n|n} = \sum_{i=1}^N \gamma_{i,n} \hat{x}_{i,n|n}$$

$$P_{n|n} = \sum_{i=1}^N \gamma_{i,n} \left(P_{i,n|n} + (\hat{x}_{n|n} - \hat{x}_{i,n|n}) (\hat{x}_{n|n} - \hat{x}_{i,n|n})^T \right).$$

Simulations in [16] show that the RPEKF outperforms the EKF under most situations, especially during initialization and low observability conditions such as very high or near zero bearing rate. This enhanced performance comes with an increase in computational complexity since the RPEKF is a collection of EKFs with the additional overhead of computing the filter weighting coefficients.

Chapter 4

Multiple Sensor Tracking with Non-Response

In this section we present a new method of fusing multiple observations in a nonlinear system while accounting for missing data. This is done by combining particle filtering with the multiple imputations (MI) technique. The MI method replaces the missing data with imputations, i.e. randomly drawn values, to form multiple complete data sets. Each data set is then particle filtered, the results of those multiple particle filtering operations are then combined as a weighted sum. The problem domain of data fusion in an asynchronous radar network is used to test the performance of this algorithm.

4.1 Problem Formulation

Consider a time-varying stochastic system with \mathbf{X}_n denoting the state at time instance n , It is assumed that \mathbf{X}_n behaves according to a non-homogenous Markov chain with transition probabilities as described by the recurrence equation

$$\mathbf{X}_n = \varphi_n(\mathbf{X}_{n-1}, W_n), \quad (4.1)$$

where W_n is random evolution noise, assumed to be an i.i.d. stochastic process and φ_n is the non-homogenous evolution transformation. Also, let the system be observed by K

sensors where the measurement is modeled

$$\mathbf{U}_n = \begin{pmatrix} \psi_1(\mathbf{X}_n, V_{n,1}) \\ \vdots \\ \psi_K(\mathbf{X}_n, V_{n,K}) \end{pmatrix} = \begin{pmatrix} U_{n,1} \\ \vdots \\ U_{n,K} \end{pmatrix},$$

let \mathbf{U}_n denotes the noisy observation of the state \mathbf{X}_n such that $V_{n,k}$ is an *i.i.d* noise process and ψ_k is the measurement transformation for sensor k , respectively. Let $U_{n,k}$ denote the k -entry of the K -dimensional vector \mathbf{U}_n . At each time instance n , some sensor observations may not be available or *missing*. In order to handle this missing data, consider the random indicator variable $\mathbf{R}_{n,k}$ which corresponds to observation $U_{n,k}$, this variable indicates if observation U_n^k is available or not

$$\mathbf{R}_{n,k} = \begin{cases} 1 & \text{observation is available from sensor } k \text{ at time } n \\ 0 & \text{observation is missing from sensor } k \text{ at time } n \end{cases}$$

Next, define the missing information set \mathbf{Z}_n as the collection of observations U_n^k at time instance n for all observers $k = 1, \dots, K$ such that $\mathbf{R}_{n,k} = 0$. Similarly, the available information set \mathbf{Y}_n is the collection of U_n^k for all $k = 1, \dots, K$ such that $\mathbf{R}_{n,k} = 1$. It is assumed that the missing data mechanism is independent of the missing observations given the available observations, this can be written as

$$P(\mathbf{R}_{n,k} | \mathbf{Z}_n, \mathbf{Y}_n) = P(\mathbf{R}_{n,k} | \mathbf{Y}_n) \text{ for all } k, n. \quad (4.2)$$

This standard statistical assumption is known as Missing at Random (MAR) [11]. Our objective is to obtain the posteriori probability density function of the state given all past and present observations, written as $p(\mathbf{X}_n | \mathbf{Y}_{0:n})$ where $\mathbf{Y}_{0:n}$ denotes all observations from the initial time instance to time instance n .

4.2 Multiple Imputations

The technique of MI was developed by Rubin [11] to deal with missing data in surveys. Missing data can introduce bias into the statistical estimation process, the existence and severity of the missing data bias depends on the missing data mechanism. Unfortunately,

the structure of the missing data mechanism is almost never available for analysis. However, under the condition of MAR (4.2), it can be shown [11] that one does not need to know the structure of the missing data mechanism. Consider K sensors all observing the same hidden state \mathbf{X} , let \mathbf{Y} denote the set of all available observations from the sensors and \mathbf{Z} denote the set of all missing observations. Moreover, let $\mathbf{R} = [\mathbf{R}_1, \dots, \mathbf{R}_K]^T$ be the K -dimensional indicator vector for the response of the sensors. Note that this problem is not time dependent and there is no state space model (4.1). The probability density $p(\mathbf{X}|\mathbf{Y}, \mathbf{R})$ can be written by the integral equation,

$$p(\mathbf{X}|\mathbf{Y}, \mathbf{R}) = \int p(\mathbf{X}|\mathbf{Y}, \mathbf{Z}, \mathbf{R}) p(\mathbf{Z}|\mathbf{Y}, \mathbf{R}) d\mathbf{Z}, \quad (4.3)$$

using the condition of MAR, it can be shown [11] that equation 4.3 reduces to

$$p(\mathbf{X}|\mathbf{Y}, \mathbf{R}) = \int p(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) p(\mathbf{Z}|\mathbf{Y}) d\mathbf{Z}.$$

This simplification implies that one does not need to know the statistical structure of the missing information mechanism. One can approximately compute the density $p(\mathbf{X}|\mathbf{Y}, \mathbf{R})$ by the Monte Carlo approximation

$$p(\mathbf{X}|\mathbf{Y}) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{j=1}^M p(\mathbf{X}|\mathbf{Y}, \mathbf{Z}_j), \quad (4.4)$$

where $\mathbf{Z}_j \sim p(\mathbf{Z}|\mathbf{Y})$ are the multiple imputations indexed by $j = 1, \dots, M$. See Figure 4.1 for a diagram of these imputations, it is easy to see that given a single partial measurement, the MI algorithm creates N likely full measurements.

Notice that MI does not use the past observations and the state transition equation in estimating the density $p(\mathbf{X}|\mathbf{Y})$. This is significant since many real world problems are well modeled by a Markov structure, which does use past values to determine the present ones. Thus, in such application, it is expected that the performance of the MI be non-optimal.

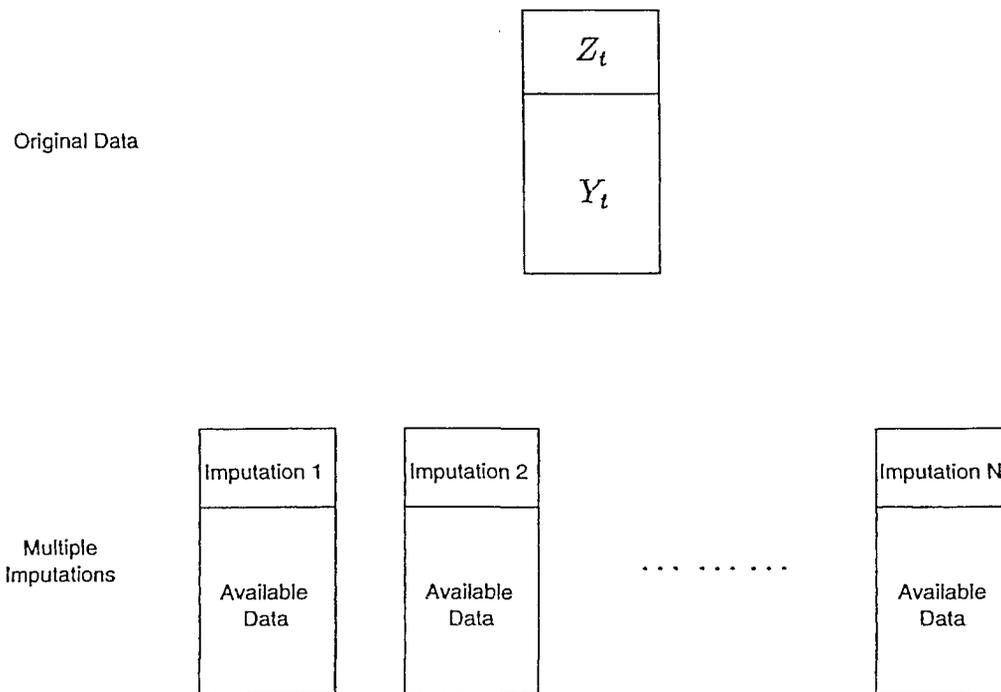


Figure 4.1: Multiple Imputation - Imputation Diagram.

4.3 Multiple Imputation Particle Filter

A new algorithm to resolves the mentioned deficiencies in the particle filtering and multiple imputations algorithms is presented. This algorithm performs the fusion using both the state and observation dynamics while accounting for the missing data.

4.3.1 Approximation of the Imputing Function

The MI method draws imputations from the missing data probability density $p(Z_n|Y_{0:n})$ as shown by Equation 4.4. However, in the context of a state space model with missing information, this density is unknown. In similar applications with unknown missing data probability density, a common solution is to draw the imputations using Markov Chain Monte Carlo (MCMC) methods [12]. These MCMC methods are iterative in nature and may not be applicable in some problem domains such as real time systems. In this section

a new approach is presented to performing the imputation process by utilizing particle approximation techniques. The imputing probability density $p(\mathbf{Z}_n|\mathbf{Y}_{0:n})$ can be written

$$p(\mathbf{Z}_n|\mathbf{Y}_{0:n}) = \int p(\mathbf{Z}_n|\mathbf{X}_n)p(\mathbf{X}_n|\mathbf{Y}_{0:n})d\mathbf{X}_n. \quad (4.5)$$

Note that the filtering density $p(\mathbf{X}_n|\mathbf{Y}_{0:n})$ appears inside the integral, since one does not know this density, $p(\mathbf{Z}_n|\mathbf{Y}_{0:n})$ cannot be sampled from directly. However, equation 4.5 suggests the following approximation. First, find a discrete density $\tilde{p}(\mathbf{X}_n|\mathbf{Y}_{0:n})$ that approximate the true filtering density well. Then using relationship 4.5, one can obtain the discrete density $\tilde{p}(\mathbf{Z}_n|\mathbf{Y}_{0:n})$, which will approximate the desired density $p(\mathbf{Z}_n|\mathbf{Y}_{0:n})$. Then one can write the approximate filtering density $\tilde{p}(\mathbf{X}_n|\mathbf{Y}_{0:n})$ using a particle approximation

$$\tilde{p}(\mathbf{X}_n|\mathbf{Y}_{0:n}) = \sum_{i=1}^{\tilde{N}} \tilde{w}_n^i \delta(\mathbf{X}_n - \tilde{\mathbf{X}}_{n,i}), \quad (4.6)$$

where the particle set $\{\tilde{w}_n^i, \tilde{\mathbf{X}}_{n,i}\}_{i=1}^{\tilde{N}}$ is obtained by performing the particle filtering with no regard for missing data. Substituting this approximation into equation 4.5 one obtains the approximate proposal function $\phi(\mathbf{Z}_n|\mathbf{Y}_{0:n})$

$$p(\mathbf{Z}_n|\mathbf{Y}_{0:n}) \approx \phi(\mathbf{Z}_n|\mathbf{Y}_{0:n}) = \sum_{i=1}^{\tilde{N}} \tilde{w}_n^i p(\mathbf{Z}_n|\tilde{\mathbf{X}}_{n,i}).$$

Assume $p(\mathbf{Z}_n|\tilde{\mathbf{X}}_{n,i}) = p(\mathbf{Y}_n|\tilde{\mathbf{X}}_{n,i})$ then this mixture of densities is known and can be sampled from in a straightforward procedure.

4.3.2 Multiple Imputations Particle Filter Algorithm

First, the filter draws random observations or imputations, from a proposal function ϕ

$$z_n^j \sim \phi(\mathbf{Z}_n|\mathbf{Y}_{0:n}) \text{ for } j = 1, \dots, M,$$

where each imputation z_n^j has an associated weight \tilde{w}_n^j as in section 4.3.1. Note that the filtering probability density $p(\mathbf{X}_n|\mathbf{Y}_{0:n})$ can be written [13]

$$p(\mathbf{X}_n|\mathbf{Y}_{0:n}) = \int p(\mathbf{X}_n|\mathbf{U}_{0:n-1}, \mathbf{Y}_n)p(\mathbf{Z}_n|\mathbf{Y}_{0:n})d\mathbf{Z}_n. \quad (4.7)$$

The Multiple Imputations Particle Filter Algorithm

- Initialize particle sets $x_0^{j,i} \sim \mu(\mathbf{X}_0)$ and set $w_0^{j,i} = 1/N$ for $i = 1, \dots, N, j = 1, \dots, M$.
 - For times $n = 1, 2, \dots$
 - Impute additional measurements $z_n^j \sim \phi(\mathbf{Z}_n | \mathbf{Y}_n)$ and set $\mathcal{U}_n^j = \{z_n^j, \mathbf{Y}_n\}$
 - For $j = 1, \dots, M$
 - Sample the importance function $x_n^{j,i} \sim \pi(\mathbf{X}_n | \mathcal{U}_n^j)$ For $i = 1, \dots, N$
 - Evaluate the importance weights: $\tilde{w}_n^i = \frac{p(\mathcal{U}_n^j | x_n^i) p(x_n^i | x_{n-1}^i)}{\pi(x_n^i | x_{0:n-1}^i, \mathcal{U}_n^j)}$ and normalize $w_n^i = \tilde{w}_n^i / \sum_{j=1}^M \tilde{w}_n^i$
For $i = 1, \dots, N$
 - For $i = 1, \dots, N$ Sample the index $d_j(i)$ distributed according to discrete **distribution** such that $\mathbb{P}(d(i) = l) = w_n^{j,l}$ for $l = 1, \dots, N$.
 - Set $x_n^{j,i} = x_n^{j,d_j(i)}$ and $w_n^{j,i} = 1/N$ for $i = 1, \dots, N$.
-

By forming the imputed data sets $\mathcal{U}_n^j = \{Z_n^j, \mathbf{Y}_n\}$ and taking a Monte Carlo approximation, rewrite equation 4.7

$$p(\mathbf{X}_n | \mathbf{Y}_{0:n}) \approx \sum_{j=1}^M \tilde{w}_n^j p(\mathbf{X}_n | \mathbf{U}_{0:n-1}, \mathcal{U}_n^j). \quad (4.8)$$

Next, the algorithm performs particle filtering on each data set \mathcal{U}_n^j to obtain the approximation

$$p(\mathbf{X}_n | \mathbf{U}_{0:n-1}, \mathcal{U}_n^j) \approx \sum_{i=1}^N w_n^{j,i} \delta(\mathbf{X}_n - x_n^{j,i}), \quad (4.9)$$

where $x_n^{j,i}$ is the i particle for the j imputation at time instance n and $w_n^{j,i}$ is its associated weight. Finally, the algorithm combines the multiple particle filtering results by substituting equation 4.9 into equation 4.8 to obtain an approximation of the desired density

$$p(\mathbf{X}_n | \mathbf{Y}_{0:n}) \approx \sum_{j=1}^M \sum_{i=1}^N \tilde{w}_n^j w_n^{j,i} \delta(\mathbf{X}_n - x_n^{j,i}).$$

See Figure 4.2 for a diagram of the full MIPF algorithm. It is clear from the diagram that the MIPF is equivalent to a bank of particle filters where each particle filter processes a single imputation. The results of each particle filter are then combined as a weighted sum to obtain a single approximation of the filtering probability density.

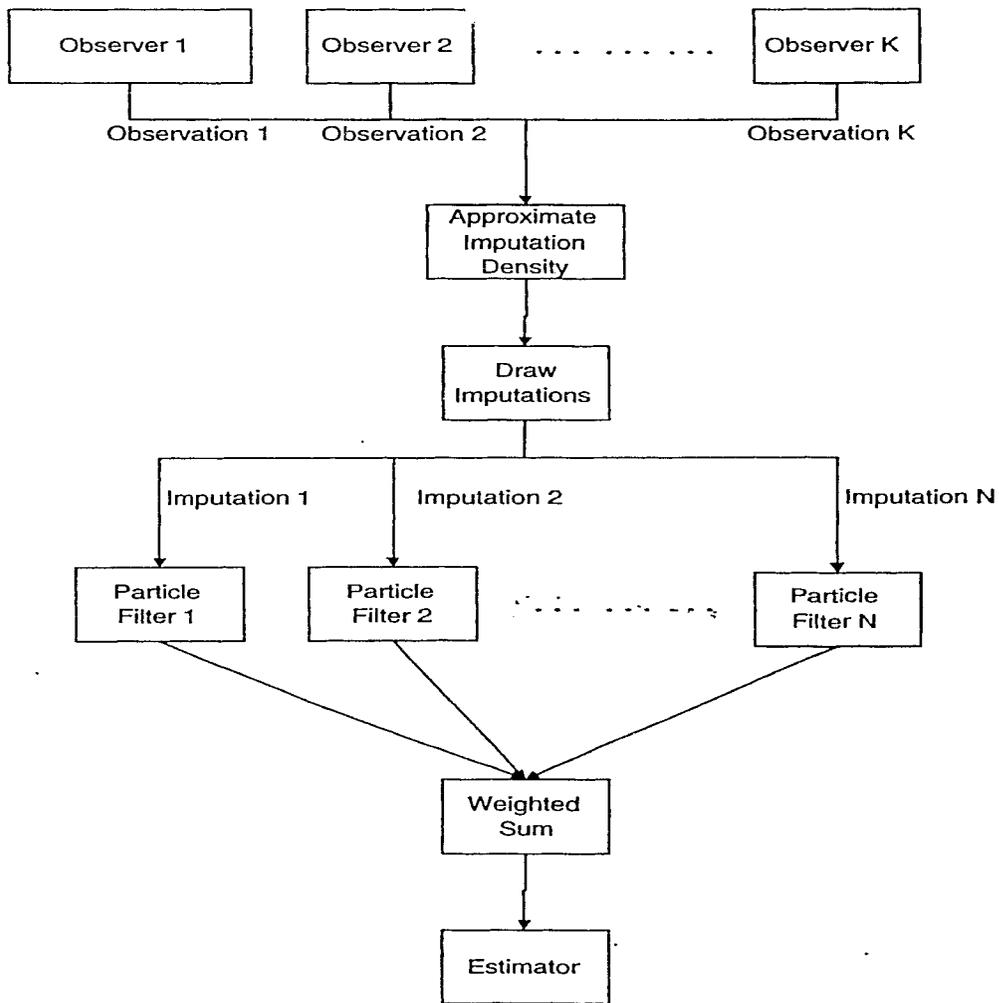


Figure 4.2: Multiple Imputation Particle Filter - Diagram.

4.4 Iterative Multiple Imputation Particle Filter

One can increase the approximation accuracy of the MIPF algorithm by introducing an iterative version of the algorithm; this modification allows us to refine the approximation of the imputation proposal function. Let m denote the iteration number of the algorithm, one can rewrite equation 4.5,

$$p_m(\mathbf{Z}_n|\mathbf{Y}_{0:n}) = \int p(\mathbf{Z}_n|\mathbf{X}_n)p_{m-1}(\mathbf{X}_n|\mathbf{Y}_{0:n})d\mathbf{X}_n. \quad (4.10)$$

where $p_m(\mathbf{Z}_n|\mathbf{Y}_{0:n})$ denotes the missing data density obtained at iteration m and $p_{m-1}(\mathbf{X}_n|\mathbf{Y}_{0:n})$ the a posteriori density function obtained at iteration $m - 1$. For the initial iteration, set $p_1(\mathbf{X}_n|\mathbf{Y}_{0:n})$ to be the approximation obtained by following the algorithm described in sections 4.3.1,4.3.2. Thus, the iterative algorithm is a feedback particle filter where the feedback data at iteration m is the estimated density $p_m(\mathbf{X}_n|\mathbf{Y}_{0:n})$.

4.5 Convergence Analysis

4.5.1 Filter Formulation in Probability Space

Let (Ω, \mathcal{F}, P) be a probability space where $\mathcal{F} = \mathcal{B}(\mathbb{R}^{n_x})$ is the Borel set of \mathbb{R}^{n_x} , the Borel set is the standard set of all possible probability events on \mathbb{R}^{n_x} . On this probability triple define a vector-valued stochastic process $X = \{X_n, n \in \mathbb{N}^+\}$ where n_x is the dimension of the state space of X . The process X is Markov with initial distribution $X_0 \sim \mu$ and probability transition kernel $K(x_n|x_{n-1})$

$$P(X_n \in A|X_{n-1} = x_{n-1}) = \int_A K(x_n|x_{n-1})dx, A \in \mathcal{B}(\mathbb{R}^{n_x}).$$

The process X can be viewed as the hidden state process to be estimated, for example in a radar tracking problem domain, X would represent be the true object position. Next, define a stochastic process $W = \{W_n, n \in \mathbb{N}^+\}$ where $W_n = \{W_n^1, \dots, W_n^k\}$ and W_n^i is an n_w -dimensional vector for $1 \leq i \leq k$. The process W is conditionally independent of X

$$P(W_n \in B|X_n = x_n) = \int_B g(w_n|x_n)dw_n, B \in \mathcal{B}(\mathbb{R}^{n_w \times k}). \quad (4.11)$$

The process W can be regarded as k noisy observations of the hidden markov process X , in our application these observations would represent all observations from multiple radars. Let the density of W conditional on X have the following factorization

$$g(w_n|x_n) = \prod_{i=1}^k g_i(w_n^i|x_n). \quad (4.12)$$

This factorization can be regarded as the requirement that given an array of k sensors, each sensor's observation is independent of all others. Thus, combining the statements 4.12 and 4.11, one arrives at

$$P(W_n^i \in C | X_n = x_n) = \int_C g(w_n^i|x_n) dw_n^i, C \in \mathcal{B}(\mathbb{R}^{n_w}).$$

Consider the non-response vector-valued stochastic process $R = \{R_n, n \in \mathbb{N}^+\}$ where R is a n_w -dimensional vector. Let $\{r_n^i \in (0, 1)\}_{i=1}^{n_w}$ be indicator variables. Define the following sets,

$$\begin{aligned} z_n &= \{w_n^i | r_n^i = 0 \text{ for } 0 \leq i \leq n_w\} \\ y_n &= \{w_n^i | r_n^i = 1 \text{ for } 0 \leq i \leq n_w\}, \end{aligned}$$

and let the probability density $h(z_n|r_{0:n}, y_{0:n})$ be

$$h(z_n|r_{0:n}, y_{0:n}) = P(Z_n \in dz_n | R_{0:m} = r_{0:m}, Y_{0:m} = y_{0:m}).$$

Clearly, the probability density h represents our knowledge of the non-response mechanism.

Consider the following probability densities of interest

$$p(x_i|y_{k:j}) = P(X_i \in dx_i | Y_k = y_k, \dots, Y_j = y_j) \quad (4.13)$$

$$p(x_m|y_{0:m}, r_{0:m}) = P(X_m \in dx_m | Y_{0:m} = y_{0:m}, R_{0:m} = r_{0:m}). \quad (4.14)$$

The probability density $p(x_m|y_{0:m}, r_{0:m})$ can be thought of as the posterior probability density which combines the data from observations and non-response while the family of probability densities $p(x_i|y_{k:j})$ describe the probability distribution of the state given only

a set of observations. Traditionally, one is interested in obtaining the probability density $p(x_n|y_n, r_n)$ to compute estimates such as MAP and MMSE. The distributions $p(x_n|y_n, r_n)$ and $p(x_n|y_n)$ are related by the following expression

$$p(x_n|y_n, r_n) = \int p(x_n|y_n) h(z_n|y_{0:n}) dz_n. \quad (4.15)$$

Standard Bayesian filtering theory gives us the equation

$$p(x_n|y_n) = \frac{g(w_n|x_n)p(x_n|y_{n-1})}{\int g(w_n|x_n)p(x_n|y_{n-1})},$$

substituting this expression into 4.15,

$$p(x_n|y_n, r_n) = \int h(z_n|w_{0:n-1}, y_n) \left(\frac{g(w_n|x_n)p(x_n|y_{n-1})}{\int g(w_n|x_n)p(x_n|y_{n-1})} \right) dz_n. \quad (4.16)$$

This equation cannot be generally solved except for very specific models such as linear Gaussian. Thus, one must resort to approximation strategies. Consider a set of values or *particles* distributed approximately according to $p(x_{n-1}|y_{n-1})$, then sample $x_n^i \sim K(x_n|x_{n-1}^i)$ using the standard bootstrap procedure. The particles are distributed approximately according to $p(x_n|y_{n-1})$ [18], they construct the empirical distributions,

$$p^N(x_n|y_{n-1}) = \frac{1}{N} \sum_{i=1}^N \delta_{x_n^i}$$

$$p^N(x_n|y_n) = \sum_{i=1}^N w_n^i \delta_{x_n^i},$$

where

$$w_n^i = \frac{g(w_n|x_n^i)}{\sum_{i=1}^N g(w_n|x_n^i)}.$$

The additional knowledge of the non-response can be incorporated into the estimation by substituting the empirical distribution $p^N(x_n|y_n)$ into Eq. 4.15 in place of the true distribution $p(x_n|y_n)$. Having done so, one arrives at the Monte Carlo approximation

$$p^N(x_n|y_n, r_n) = \int h(z_n|w_{0:n-1}, y_n) p^N(x_n|y_n) dz_n.$$

The integral above cannot be evaluated explicitly so a naive Monte Carlo procedure is adopted to approximate the integral. Let the imputation set $\{z^j\}_{j=0}^M$ be sampled from the nonresponse density, $z_n^j \sim h(z_n|w_{0:n-1}, y_n)$, then one arrives at the relation

$$p^{N,M}(x_n|y_n, r_n) = p^N(x_n|y_n, r_n) \frac{1}{M} \sum_{j=1}^M \delta_{z_n^j}.$$

Therefore, the empirical density $p^{N,M}(x_n|y_n, r_n)$ approximates the desired probability density $p(x_n|y_n, r_n)$ in terms of two sets $\{w_y^i, x_y^i\}_{i=1}^N$ and $\{z_n^j\}_{j=1}^M$.

4.5.2 Almost Sure Convergence

The integral expression 4.16 can be thought of as a sequence of three transformations whose overall result is taking a probability density $p^{N,M}(x_{n-1}|y_{n-1}, r_{n-1})$ to the next one in time $p(x_n|y_n, r_n)$. One can write this sequence of maps as $p(x_n|y_{n-1}) \rightarrow p(x_n|y_n) \rightarrow p(x_n|y_n, r_n)$ where $p(x_n|y_{n-1})$ and $p(x_n|y_n)$ are intermediate distributions due to the particle filtering. In order to prove the convergence of this sequence of mappings, let us begin with an abstract argument and later on show how it is related to the algorithm at hand. Consider a metric space (E, d) and let $\{a_n\}_{n=1}^\infty$, $\{b_n\}_{n=1}^\infty$ and $\{d_n\}_{n=1}^\infty$ be sequences of continuous functions $a_n, b_n, d_n : E \rightarrow E$ indexed by $n \in \mathbb{N}^+$. Also, let

$$l_n = d_n \circ a_n \circ b_n.$$

Define two, not necessarily continuous, perturbation functions $f^M, c^N : E \rightarrow E$ in the following way. Let us assume that as N and M increase, c^N and f^M will converge to the identity function. Now, perturb k_n using these two functions

$$l_n^{N,M} = f^M \circ d_n \circ c^N \circ a_n \circ c^N \circ b_n.$$

Let e_M and e_N be a sequence of elements in the metric space E indexed by M and N respectively and let $e \in E$ denote a single element of E . It is assumed that f^M, c^N satisfy the following conditions for all such sequences e_M, e_N ,

$$\lim_{N \rightarrow \infty} e_N = e \Rightarrow \lim_{N \rightarrow \infty} c^N(e_N) = e \tag{4.17}$$

$$\lim_{M \rightarrow \infty} e_M = e \Rightarrow \lim_{M \rightarrow \infty} f^M(e_M) = e, \tag{4.18}$$

then the following lemma can be shown true [18]

Lemma 3 Let a_n, b_n, k_n and c^N be as defined above. Then if c^N satisfies condition 4.17,

$$\lim_{N \rightarrow \infty} k_n^N = k_n.$$

Moreover, k_n^N satisfies

$$\lim_{N \rightarrow \infty} e_N = e \Rightarrow \lim_{N \rightarrow \infty} k_n^N(e_N) = k_n(e).$$

Then one can prove the following lemma

Lemma 4 Let d_n, a_n, b_n and f^M, c^N be as defined above, then if conditions 4.17-4.18 are satisfied,

$$\lim_{N \rightarrow \infty} e_N = e \Rightarrow \lim_{M \rightarrow \infty} f_M(\lim_{N \rightarrow \infty} (d_n \circ k_n^N)(e_N)) = l_n(e).$$

Proof Let $l_n^{N,M} = f^M \circ d_n \circ k_n^N$ where $k_n^N = c^N \circ a_n \circ c^N \circ b_n$ and $k_n = a_n \circ b_n$. Then by Lemma 3,

$$\lim_{N \rightarrow \infty} e_N = e \Rightarrow \lim_{N \rightarrow \infty} k_n^N(e_N) = k_n(e),$$

since d_n is continuous

$$\lim_{N \rightarrow \infty} k_n^N(e_N) = k_n(e) \Rightarrow \lim_{N \rightarrow \infty} (d_n \circ k_n^N)(e_N) = (d_n \circ k_n)(e).$$

Now set $\widehat{e} = (d_n \circ k_n)(e)$ and $\widehat{e}_M = \lim_{N \rightarrow \infty} (d_n \circ k_n^N)(e_N)$ for all M then clearly, $\widehat{e}, \widehat{e}_M \in E$ and $\lim_{M \rightarrow \infty} \widehat{e}_M = \widehat{e}$. Using condition 4.18, the argument follows

$$\begin{aligned} \lim_{M \rightarrow \infty} \widehat{e}_M = \widehat{e} &\Rightarrow \lim_{M \rightarrow \infty} f_M(\widehat{e}_M) = \widehat{e} \\ &\Rightarrow \lim_{M \rightarrow \infty} f_M(\lim_{N \rightarrow \infty} (d_n \circ k_n^N)(e_N)) = (d_n \circ k_n)(e) = l_n(e). \end{aligned}$$

■

Specializing this abstract discussion to the domain of MIPF, let $E = \mathcal{P}^{n_x}$ be the space of probability measures endowed with the topology of weak convergence as described in

section 2.1.3. Let $\nu \in \mathcal{P}^{\mathbb{R}^x}$ be an arbitrary probability measure and φ is any continuous bounded function, define the mappings

$$\begin{aligned} (a_n(\nu), \varphi) &= \left(\int_{\mathbb{R}^{\mathbb{R}^x}} g(y_n|x_n) \nu(dx_n) \right)^{-1} \int_{\mathbb{R}^{\mathbb{R}^x}} \varphi(x_n) g(y_n|x_n) \nu(dx_n) \\ (b_n(\nu), \varphi) &= \int_{\mathbb{R}^{\mathbb{R}^x}} \int_{\mathbb{R}^{\mathbb{R}^x}} \varphi(x_n) K(dx_n|x_{n-1}) \nu(dx_{n-1}). \end{aligned}$$

Assuming K is Feller (i.e. and the function g is continuous bounded strictly positive then a_n, b_n can be shown to be continuous. Also, let the sequence of functions d_n be mappings $d_n : \mathcal{P}^{\mathbb{R}^x} \rightarrow \mathcal{P}^{\mathbb{R}^x}$ such that

$$d_n(\nu) = \int_{\mathbb{R}^{\mathbb{R}^z}} \nu(dx_n|z_n, y_n) H(dz_n), \quad (4.19)$$

which implies

$$\begin{aligned} (d_n(\nu), \varphi) &= \int_{\mathbb{R}^{\mathbb{R}^z}} \int_{\mathbb{R}^{\mathbb{R}^x}} \varphi(x_n) \nu(dx_n|z_n, y_n) H(dz_n) \\ &= (H, \nu\varphi). \end{aligned}$$

It is convenient for the mappings d_n to be a continuous operator, this is quite reasonable since one can interpret that requirement as the fact that adding or removing observers will influence the observations in a continuous manner (where the continuity is in the function space sense). Moreover, define the perturbation functions c^N and f^M

$$\begin{aligned} c^N &= \frac{1}{N} \sum_{i=1}^N \delta_{V_i} \\ f^M &= \frac{1}{M} \sum_{j=1}^M \delta_{W_j}, \end{aligned}$$

where V_i, W_j are *i.i.d.* random variables with common distributions ν_c, ν_f respectively. One arrives at the following lemma,

Lemma 5 *If c^N and f^M are as defined above, then they satisfy conditions 4.17-4.18 almost surely.*

Consider the empirical measure $p^{N,M}(x_n|y_n, r_n)$, it is easy to see that

$$p^{N,M}(x_n|y_n, r_n) = f^M \circ d_n \circ c^N \circ a_n \circ c^N \circ b_n(p^{N,M}(x_{n-1}|y_{n-1}, r_{n-1})) = k_n^{N,M}(p^{N,M}(x_{n-1}|y_{n-1}, r_{n-1})).$$

Thus, one obtains the following theorem:

Theorem 2 *Assuming the transition kernel K is Feller and g is bounded, continuous and strictly positive, then almost surely*

$$\lim_{N \rightarrow \infty} \lim_{M \rightarrow \infty} p^{N,M}(x_n|y_n, r_n) = p(x_n|y_n, r_n),$$

where the convergence is in the weak sense

Proof Using the definitions above,

$$p^N(x_n|y_n) = (c^N \circ a_n \circ c^N \circ b_n)(p^N(x_{n-1}|y_{n-1})) = k_n^N(p^N(x_{n-1}|y_{n-1})),$$

and if $\lim_{N \rightarrow \infty} p^N(x_{n-1}|y_{n-1}) = p(x_{n-1}|y_{n-1})$ then $\lim_{N \rightarrow \infty} p^N(x_n|y_n) = p(x_n|y_n)$. Also, note that

$$p^{N,M}(x_n|y_n, r_n) = (f^M \circ d_n \circ k_n^N)(p^N(x_{n-1}|y_{n-1})),$$

then by lemmas 4 and 5, the argument follows

$$\lim_{M \rightarrow \infty} (f^M \lim_{N \rightarrow \infty} (d_n \circ k_n^N)(p^N(x_{n-1}|y_{n-1}))) = d_n \circ a_n \circ b_n = p(x_n|y_n, r_n),$$

which implies

$$\lim_{N \rightarrow \infty} \lim_{M \rightarrow \infty} p^{N,M}(x_n|y_n, r_n) = p(x_n|y_n, r_n). \quad \blacksquare$$

The convergence theorem 2 above indicates that the MIPF will convergence in the weak sense with probability 1 (almost surely) as the number of particles approaches infinity. However, this result does not guarantee that this convergence will happen with a finite number of particles nor that the squared error will be bounded.

4.6 Performance Analysis

4.6.1 Radar Network Data Fusion

Many radar systems are implemented using a radar network in order to cover a large area and increase reliability, these radars transmit observations asynchronously due to the

large distances between the radars. The radar network's state space model is formulated and then simulated in order to examine the performance of the MIPF as compared against the EM algorithm.

Multiple Radars Model

Consider K radars measuring a common target. The target motion is described by a simple near constant velocity state model [1]

$$X_{n+1} = \Phi_{n+1}X_n + \Gamma_{n+1}V_n,$$

the matrices Φ_{n+1} and Γ_{n+1} have been defined in section 3.1. The state vector X_n is defined as $X_n = [x(n), v_x(n), y(n), v_y(n)]^T$, and the system noise $V_n \sim \mathcal{N}(0, \Sigma_v)$ is assumed to be a Gaussian *i.i.d.* process. Let φ denote the cartesian to polar transformation

$$\psi(X_n) = \begin{pmatrix} \sqrt{x(n)^2 + y(n)^2} \\ \tan^{-1} \left(\frac{y}{x} \right) \end{pmatrix}.$$

Also let the radar observation model be given by

$$Y_n^j = \varphi(X_n) + W_n^j,$$

where $W_n \sim \mathcal{N}(0, \Sigma_w^j)$ denotes the system noise, which is assumed to be a Gaussian *i.i.d.* process. It is also assumed that the system and observation noises are mutually independent.

4.6.2 Simulation Results

Two radar network configurations are simulated: a network of two asynchronous radars and a network of three asynchronous radars. The performance of the MIPF is compared against the standard EM fusion algorithm [12]. The two radar network is simulated for twenty time units where at each time instance, the observations are repeated thirty times to obtain performance results. Similarly, the three radar network is simulated for twelve time units where at each time instance, the observations are repeated ten times. Both

simulation configurations use a particle filter with $N = 250$ and $M = 40$. The performance results for the two radar network are shown in Figure 4.3, the error metric is mean squared error. It is clear that the MIPF offers significant performance improvement over the EM algorithm. Figure 4.4 shows some performance gains by the MIPF for the three radar configuration. However, the performance gain is not as significant as for the two radar network. This discrepancy can be explained as follows, one expects that the more dominant the missing data behavior of the system, the more performance gains we will realize from the multiple imputation particle filter, which is designed to deal with missing data, as compared to the EM algorithm. Due to the nature of the simulation, the ratio of missing to available information for the simulated two radar network is 50% while the missing information ratio for the simulated three radar network is almost always 25%. Then, as expected, the multiple imputation particle filter shows significant performance gain for the more aggressive missing data pattern of the two radar network.

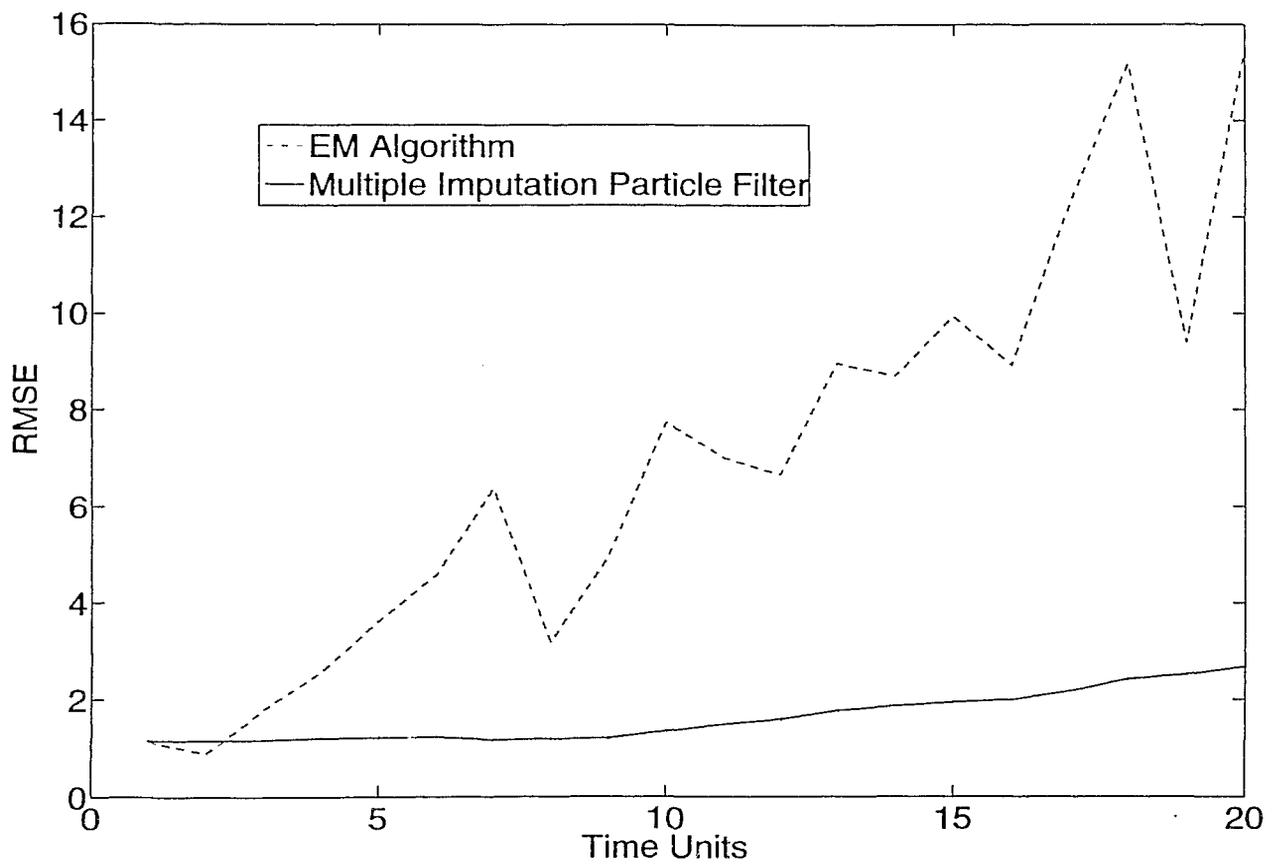


Figure 4.3: Performance of MIPF and EM for Two Radars - Missing Data Ratio of %50

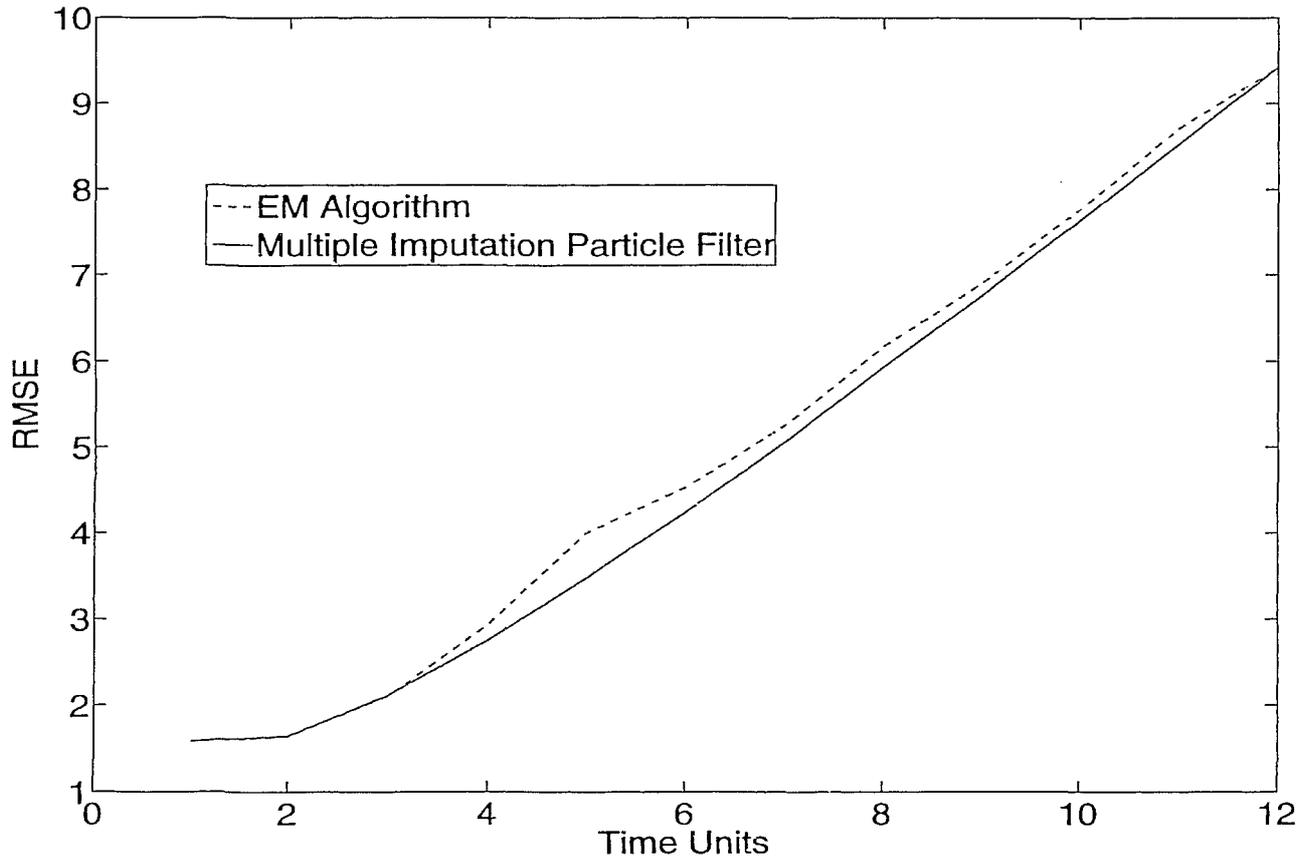


Figure 4.4: Performance of MIPF and EM for Three Radars - Missing Data Ratio of %25

Chapter 5

Sequential Monte Carlo Methods for Multi-Site Bearing Only Tracking

Bearing-only tracking is a highly nonlinear problem that can be effectively tackled using SMC methods presented in Chapters 2 and 3. However, a direct application of the particle filter to the bearing only problem is problematic since incorrect filter initialization can cause divergent behavior. A new method for particle filter initialization is presented in this chapter, this approach is applicable for the situation of multiple bearing only sensors. It operates by resolving the initial state using a least squares approach.

5.1 Problem Formulation

Consider a target moving in two dimensional space, for simplicity we'll assume that the target does not maneuver and follows a near constant velocity model [1]. Thus, one has state dynamics equation

$$\mathbf{X}_n = \Phi_n \mathbf{X}_{n-1} + \Gamma_n \mathbf{V}_n, \quad (5.1)$$

the matrices Φ_n and Γ_n have been defined in section 3.1, the state vector is written $\mathbf{X}_n = [x(n), y(n), v_x(n), v_y(n)]^T$ and $\mathbf{V}_n \sim \mathcal{N}(0, \Sigma_v)$ is the state transition noise. Assume that K sensors are taking noisy bearing measurements of the target. These sensors are modeled

using the standard bearing only measurement model

$$\mathbf{Y}_n = \begin{pmatrix} Y_n^1 \\ Y_n^2 \\ \vdots \\ Y_n^K \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{y_n - y_{s,n}^1}{x_n - x_{s,n}^1}\right) \\ \arctan\left(\frac{y_n - y_{s,n}^2}{x_n - x_{s,n}^2}\right) \\ \vdots \\ \arctan\left(\frac{y_n - y_{s,n}^K}{x_n - x_{s,n}^K}\right) \end{pmatrix} + \mathbf{W}_n. \quad (5.2)$$

Let the symbols $y_{s,n}^i$, $x_{s,n}^i$ denote the cartesian position of the i -th sensor at time n . The observation noise is a K -dimensional Gaussian noise distribution, $\mathbf{W}_n \sim \mathcal{N}(\mathbf{M}_w, \Sigma_w)$.

5.2 Least Squares Initialization

As discussed in the previous section, tracker initialization is a major consideration in any bearing only tracking systems. Therefore, a new initialization technique is proposed; it uses previous work done by Don Koks [28] and combines it with particle filtering techniques. Consider the probability distribution $\mu(\mathbf{X}_0)$ which describes the statistical behavior of the target's initial state and assume that its initial velocity components are known. Therefore, one needs to generate the particles' position entries using the initial bearing measurements \mathbf{Y}_0 . The idea is to apply a least squares procedure to obtain multiple estimates of the target position. Next, assume the initial distribution is Gaussian and estimate its statistics using the multiple estimates. Finally, one can draw the particles from this initial distribution.

Following [28], write the initial range and bearing of the target as r_i and θ_i respectively, where the index i denotes that these range and bearing are taken by the i -th sensor. The following trigonometric relations are true,

$$\begin{aligned} r_i \cos \theta_i &= x_0 - x_{s,0}^i \\ r_i \sin \theta_i &= y_0 - y_{s,0}^i, \end{aligned}$$

for all sensors $1 \leq i \leq K$. By writing these relations for all sensors, one can eliminate the unknown initial coordinates $[x_0, y_0]$ and rewrite the trigonometric equations in matrix

form,

$$\mathbf{A}[r_1 r_2 \cdots r_K]^T = \mathbf{B}, \quad (5.3)$$

where

$$\mathbf{A} = \begin{pmatrix} \cos \theta_1 & -\cos \theta_2 & 0 & 0 & \cdots & 0 \\ \sin \theta_1 & -\sin \theta_2 & 0 & 0 & \cdots & 0 \\ 0 & \cos \theta_2 & -\cos \theta_3 & 0 & \cdots & 0 \\ 0 & \sin \theta_2 & -\sin \theta_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cos \theta_{K-1} & -\cos \theta_K \\ 0 & 0 & \cdots & \sin \theta_{K-1} & -\sin \theta_K \end{pmatrix}, \quad (5.4)$$

and

$$\mathbf{B} = \begin{pmatrix} -x_{s,0}^1 + -x_{s,0}^2 \\ -y_{s,0}^1 + -y_{s,0}^2 \\ -x_{s,0}^2 + -x_{s,0}^3 \\ -y_{s,0}^2 + -y_{s,0}^3 \\ \vdots \\ -x_{s,0}^{K-1} + -x_{s,0}^K \\ -y_{s,0}^{K-1} + -y_{s,0}^K \end{pmatrix}. \quad (5.5)$$

This is an over-determined matrix equation since \mathbf{A} is not a square matrix, a single solution does not exist so one may opt for the standard least squares solution which minimizes the squared error. Let the symbol \bar{r}_i denote the range estimate at the coordinate system of the i -th sensor, then write the least squares solution as,

$$[\bar{r}_1, \bar{r}_2 \cdots \bar{r}_K]^T = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B}. \quad (5.6)$$

Transferring these estimates to the target's coordinate system,

$$\begin{aligned} \bar{\mathbf{X}}_1 &= (\bar{r}_1 \cos \theta_1 + x_{0,s}^1, \bar{r}_1 \sin \theta_1 + y_{0,s}^1)^T \\ \bar{\mathbf{X}}_2 &= (\bar{r}_2 \cos \theta_2 + x_{0,s}^2, \bar{r}_2 \sin \theta_2 + y_{0,s}^2)^T \\ &\vdots \\ \bar{\mathbf{X}}_K &= (\bar{r}_K \cos \theta_K + x_{0,s}^K, \bar{r}_K \sin \theta_K + y_{0,s}^K)^T \end{aligned} \quad (5.7)$$

Therefore, there are K estimates of the initial position of the initial target. One can com-

pute the sample mean and variance of these estimates,

$$\mathbf{M}_\mu = \frac{1}{K} \sum_{j=1}^K \bar{\mathbf{X}}_j \quad (5.8)$$

$$\Sigma_\mu = \frac{1}{K-1} \sum_{j=1}^K (\bar{\mathbf{X}}_j - \mathbf{M}_\mu) (\bar{\mathbf{X}}_j - \mathbf{M}_\mu)^T. \quad (5.9)$$

Let us make the assumption that the initial state distribution $\mu(\mathbf{X}_0)$ is Gaussian, since the initial distribution is Gaussian, one can use the sample mean (Eq. 5.8) and variance (Eq. 5.9) to estimate the distribution's parameters and let $\mu(\mathbf{X}_0) \sim \mathcal{N}(\mathbf{M}_\mu, \Sigma_\mu)$. Finally, we've obtained an approximate form of the initial position's probability density function.

5.3 Bearing only Target tracking with Monte Carlo Algorithms

As discussed in section 2.4, one is interested in approximating the probability density distribution $p(\mathbf{X}_n | \mathbf{Y}_n)$, this is done by using a set of particles $\{\mathbf{x}_n^j\}_{j=1}^N$ and their associated weights $\{w_n^j\}_{j=1}^N$ where N is the number of particles and j indexes the particle set. The particles are initialized by sampling from the initial state distribution $\mu(\mathbf{X}_0)$,

$$\begin{aligned} \mathbf{x}_0^i &\sim \mathcal{N}(\mathbf{M}_\mu, \Sigma_\mu) \\ w_0^i &= 1/N, \end{aligned}$$

where the initial distribution $\mu(\mathbf{X}_0)$ is obtained by the least squares procedure. At every time iteration, the particles are sampled from an importance function,

$$\mathbf{x}_n^j \sim \pi(\mathbf{X}_n | \mathbf{x}_{n-1}^j, \mathbf{Y}_n), \quad (5.10)$$

where $\pi(\mathbf{X}_n | \mathbf{x}_{n-1}^j, \mathbf{Y}_n)$ denotes the importance function that is yet to be specified. A particle's weight is evaluated up to a normalizing constant by the following recursive formula

$$w_n^j = w_{n-1}^j \frac{p(\mathbf{Y}_n | \mathbf{x}_{n-1}^j) p(\mathbf{x}_n^j | \mathbf{x}_{n-1}^j)}{\pi(\mathbf{X}_n | \mathbf{x}_{n-1}^j, \mathbf{Y}_n)}. \quad (5.11)$$

The coefficients are then normalized by dividing the given coefficient by the sum of all weighting coefficients.

5.3.1 Particle Filter for Bearing Only Tracking

A straight forward application of the particle filter algorithm to the bearing only tracking application uses the prior importance function (See section 2.4 for more information) and computes the weight using the likelihood function. The prior importance function $p(\mathbf{X}_n|\mathbf{X}_{n-1})$ can be written

$$p(\mathbf{X}_n|\mathbf{x}_{n-1}^j) = \frac{1}{4\pi^2|\Gamma_n^T \Sigma_v \Gamma_n|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{X}_n - \Phi_n \mathbf{x}_{n-1}^j)^T (\Gamma_n^T \Sigma_v \Gamma_n)^{-1} (\mathbf{X}_n - \Phi_n \mathbf{x}_{n-1}^j) \right\}, \quad (5.12)$$

and weighting formula

$$w_n^j = w_{n-1}^j p(\mathbf{Y}_n|\mathbf{x}_{n-1}^j), \quad (5.13)$$

where the likelihood function $p(\mathbf{Y}_n|\mathbf{x}_{n-1}^j)$ is

$$p(\mathbf{Y}_n|\mathbf{x}_{n-1}^j) \sim \mathcal{N}(\arctan(\mathbf{x}_{n-1}^j), \Sigma_w). \quad (5.14)$$

As noted in chapter 2, the prior importance function does not use any measurement information in drawing the particle and thus may require more particles as compared to the optimal importance function. In order to incorporate the measurement into the importance function, let us first linearize the state space model and then derive its optimal importance function. Such an approach to importance functions is known as the *linearized optimal importance function* (See [8] and section 2.4), note that this idea is conceptually similar to the EKF. Following the derivation in 2.4, let the function $g : \mathbb{R}^4 \rightarrow \mathbb{R}^K$ denote the observation transformation (Eq. 5.2) and $f : \mathbb{R}^4 \rightarrow \mathbb{R}^4$ denote the state transition equation (Eq. 5.1)

$$\mathbf{X}_n = f(\mathbf{X}_{n-1}) + \mathbf{V}_n \quad (5.15)$$

$$\mathbf{Y}_n = g(\mathbf{X}_n) + \mathbf{W}_n, \quad (5.16)$$

then the linearized state space model's observation equation is written

$$\mathbf{Y}_n \simeq g(f(\mathbf{X}_{n-1})) + \frac{\partial g(\mathbf{X}_n)}{\partial \mathbf{X}_n} (\mathbf{X}_n - f(\mathbf{X}_{n-1})). \quad (5.17)$$

Calculating the measurement equation's Jacobian,

$$\frac{\partial g(\mathbf{X}_n)}{\partial \mathbf{X}_n} = \Lambda = \begin{pmatrix} \frac{(-y+y_{s,n}^1)}{(x-x_{s,n}^1)^2+(y-y_{s,n}^1)^2} & \frac{1}{(x-x_{s,n}^1)+\frac{(y-y_{s,n}^1)^2}{(x-x_{s,n}^1)}} & 0 & 0 \\ \frac{(-y+y_{s,n}^2)}{(x-x_{s,n}^2)^2+(y-y_{s,n}^2)^2} & \frac{1}{(x-x_{s,n}^2)+\frac{(y-y_{s,n}^2)^2}{(x-x_{s,n}^2)}} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{(-y+y_{s,n}^K)}{(x-x_{s,n}^K)^2+(y-y_{s,n}^K)^2} & \frac{1}{(x-x_{s,n}^K)+\frac{(y-y_{s,n}^K)^2}{(x-x_{s,n}^K)}} & 0 & 0 \end{pmatrix}. \quad (5.18)$$

Then the **optimal** importance function of the linearized state space model consisting of equations 5.17 and 5.1 can be written as

$$\pi(\mathbf{X}_n | \mathbf{x}_{n-1}^j, \mathbf{Y}_n) \sim \mathcal{N}(\mathbf{M}_n, \Sigma_n), \quad (5.19)$$

where

$$\Sigma_n^{-1} = (\Gamma_n^T \Sigma_v \Gamma_n)^{-1} + \Lambda^T \Sigma_w^{-1} \Lambda \quad (5.20)$$

$$\mathbf{M}_n = \Sigma_n \left((\Gamma_n^T \Sigma_v \Gamma_n)^{-1} f(\mathbf{X}_{n-1}) + \Lambda^T \Sigma_w^{-1} \times (\mathbf{Y}_n - g(f(\mathbf{X}_{n-1})) + \Lambda f(\mathbf{X}_{n-1})) \right) \quad (5.21)$$

Using this importance function, the full algorithm can be formed by combining the particle filter derived above with the least squares initialization scheme.

5.4 Performance Analysis

5.4.1 Simulation Results

The simulation scenario consists of three bearing only observation stations, each one observing the same target. The target dynamics obey the state space model (5.1) described in the beginning of this chapter. Two tracking algorithms are compared, the EKF and the particle filter. The EKF's initial state is initialized with the ground truth position and its initial covariance matrix is the zero matrix. Thus the EKF will have no divergent behavior associated with an incorrect initialization. In contrast, the particle filter is initialized using the least squares initialization technique described in section 5.2. Figure 5.1 shows

Bearing Only Particle Filter Algorithm with Least Squares Initialization

- Compute M_μ, Σ_μ and initialize the particle set $x_0^i \sim \mathcal{N}(M_\mu, \Sigma_\mu)$, let $w_0^i = 1/N$ for $i = 1, \dots, N$.
 - For times $n = 1, 2, \dots$
 - Sample the importance function $x_n^i \sim \mathcal{N}(M_n, \Sigma_n)$ For $i = 1, \dots, N$
 - Evaluate the importance weights: $\tilde{w}_n^i = \frac{p(Y_n|x_n^i)p(x_n^i|x_{n-1}^i)}{\pi(x_n^i|x_{0:n-1}^i, Y_{0:n})}$ and normalize $w_n^i = \tilde{w}_n^i / \sum_{j=1}^N \tilde{w}_n^j$
For $i = 1, \dots, N$
 - For $i = 1, \dots, N$ Sample the index $d(i)$ distributed according to discrete distribution such that $\mathbb{P}(d(i) = l) = w_n^l$ for $l = 1, \dots, N$.
 - Set $x_n^i = x_n^{d(i)}$ and $w_n^i = 1/N$ for $i = 1, \dots, N$.
-

the mean square error performance associated with the EKF and particle filter. From Figure 5.1, one notes that the particle filter's initial estimation performance is worse than the EKF since the EKF uses the ground truth for initialization while the particle filter uses the least squares approach. However, the crucial observation is that the particle filter's performance improves after a few time iterations and does not diverge. Indeed, the particle filter's performance exceeds that of the EKF in spite of the EKF's initialization advantage. Additional simulations were performed using the same sensor configuration. These simulations were performed by generating 50 Monte Carlo tracks with identical initial state and model parameters. Each track was then recursively filtered using both an EKF and a particle filter. The root mean square error (RMSE) between each track's ground truth and estimated position is computed. The errors of all tracks at a given time are averaged to produce an overall initial performance graph which is shown in Figure 5.2. The first tracker is a particle filter initialized using the new scheme developed above. This algorithm is compared against an extended Kalman filter which is initialized with the true initial position of the target. In other words, the EKF's initialization is optimal while the particle filter uses the non-perfect initialization. It is clear from Figure 5.2 that on average,

the initialization of the particle filter produces the same initial performance as that of the EKF initialized with the true state. These simulation results allow one to conclude that the initialization technique is successful since it enables the particle filter to rapidly converge to its optimal estimation without any loss of initial performance.

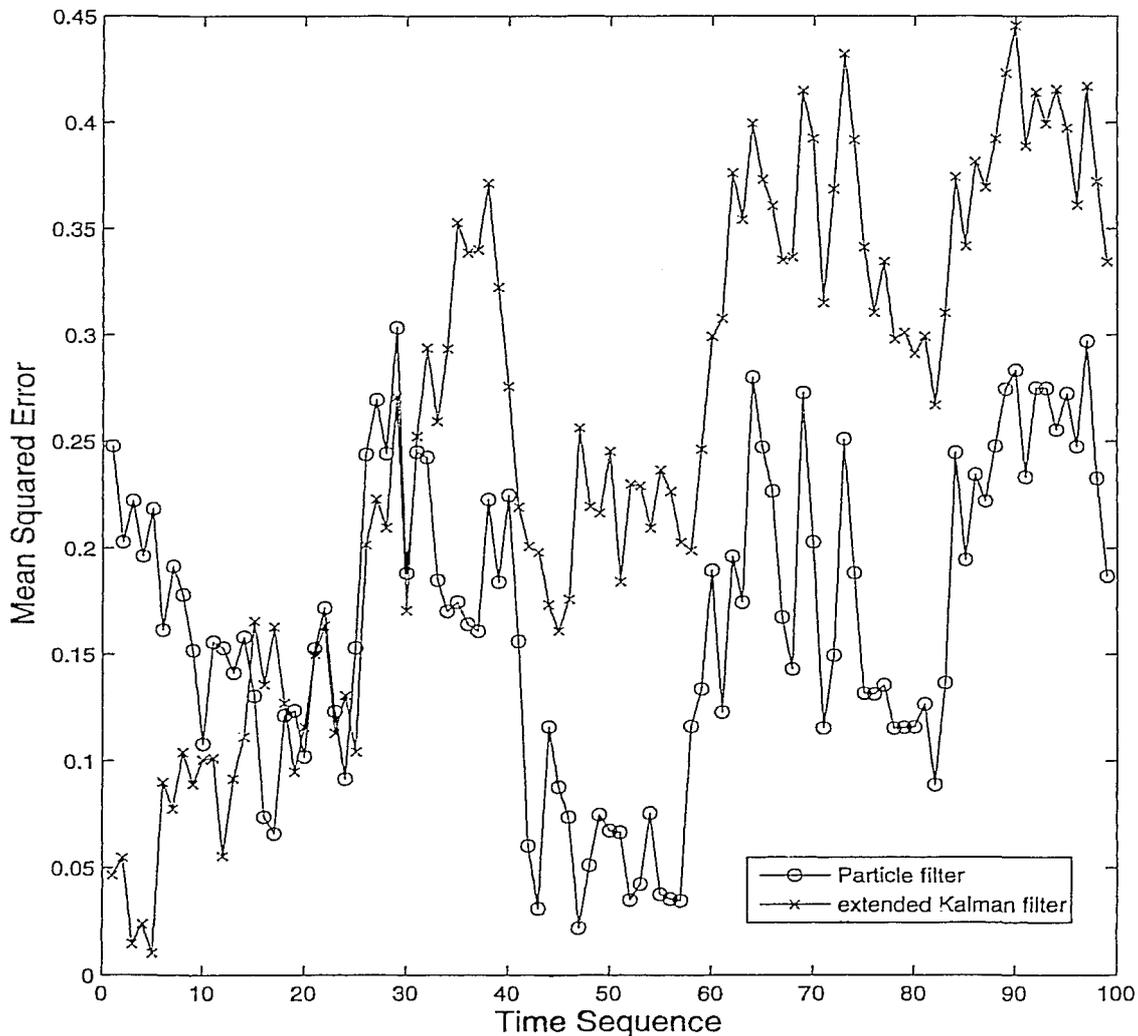


Figure 5.1: Performance of the EKF and Particle filter in Bearing Only Tracking.

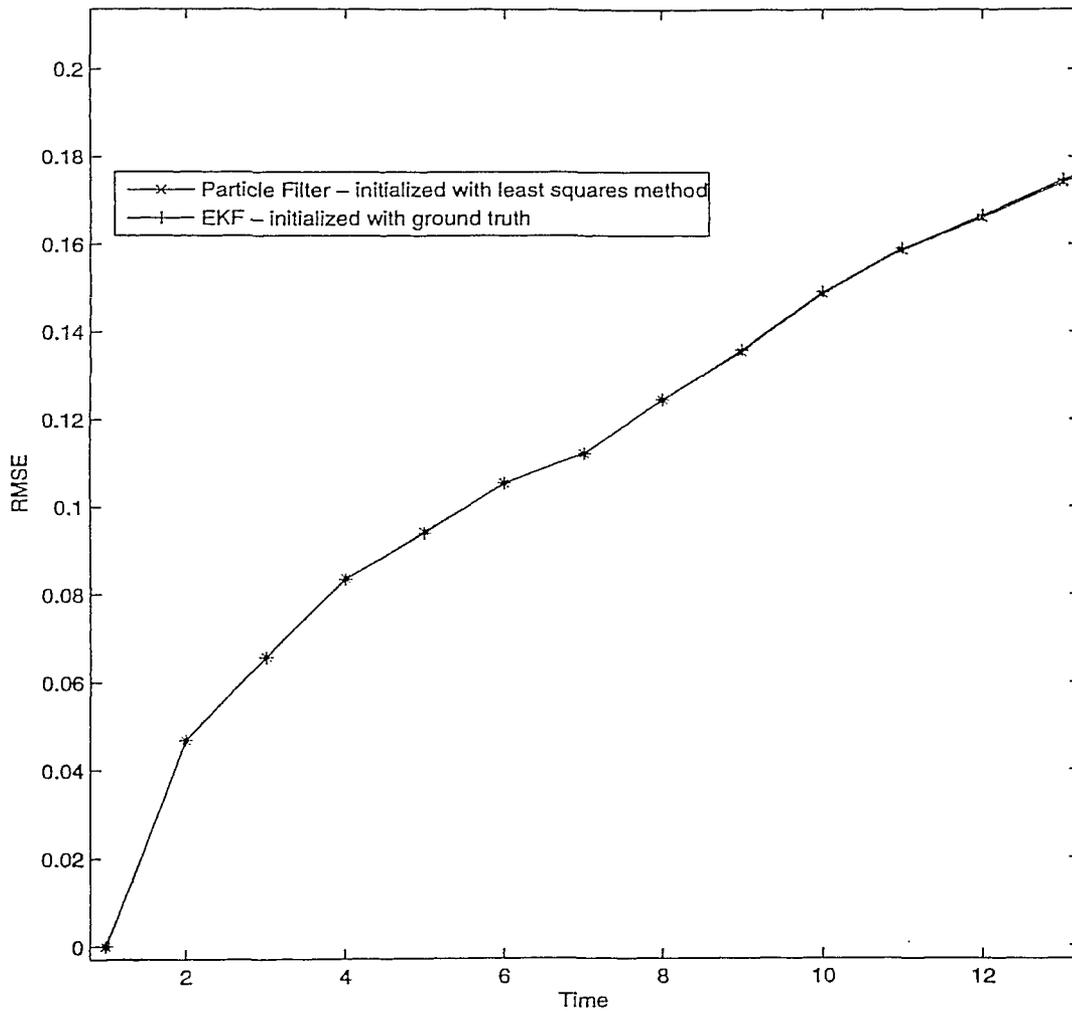


Figure 5.2: Average performance of the EKF and Particle filter in Bearing Only Tracking.

Chapter 6

Concluding Remarks

6.1 Conclusions

The work presented largely deals with the application of Monte Carlo methods to the problem of multi-site radar and bearing only target tracking. While radar and bearing only tracking are well studied problems, the particle filtering framework introduces new flexibilities and issues that need to be resolved. First, the problem of target tracking in asynchronous sensor networks is considered. This problem is identified as the problem of fusing multiple sensor measurements in the presence of missing data. The MIPF is introduced to systematically deal with such problems of estimation with missing data. The MIPF algorithm draws imputations using a new particle based method, it then combines the multiple imputations with a particle filtering technique. Some convergence properties are derived for the MIPF, these theoretical results suggest that the algorithm converges in a desirable manner. Simulated data is used to demonstrate the effectiveness and performance of the proposed algorithm. Next, a new multi-site bearing only initialization scheme for a particle filter-based tracking algorithm is proposed. The initialization schemes utilizes a least squares approach to resolve the initial state of the target. The particle filter is tested with the initialization scheme using simulated data and compared against a standard EKF. The initialization scheme is shown to perform well and allows the Monte Carlo algorithm to rapidly converge to the optimal filter.

6.2 Future Research

The weak convergence of the Multiple Imputations Particle filter has been studied in Chapter 4. Additional research is required to explore whatever the algorithm's error is bounded and in what sense. Also, the utilization of asynchronous measurements in enhancing the performance of model learning and system identification deserves additional study, especially in the context of particle based system identification methods. Indeed, model learning schemes are a major consideration for all mode based tracking algorithms such as the particle filter. In much of the tracking literature, tracker initialization is relegated to the realm of the practitioner, however the research in this thesis demonstrates that filter initialization is a significant problem that bears some relation to model selection and learning. Therefore, initialization schemes and their theoretical effect on the stability of sequential Monte Carlo algorithms require additional research, especially in the context of filters that use a slightly inaccurate model in their derivation.

Bibliography

- [1] Y. Zhou, A Kalman Filter based Registration Approach for Multiple Asynchronous Sensors, Defense R&D Canada - Ottawa, 2003.
- [2] P. L. Bogler, Radar Principles with Applications to Tracking Systems, Wiley, New York, 1990.
- [3] B. Anderson and J. Moore. Optimal Filtering. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [4] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series," J. Amer. Statist. Assoc., vol. 82, pp. 1032–1041, Dec. 1987.
- [5] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," IEE Proceedings-F, vol. 140, pp. 107–113, April 1993.
- [6] G. Kitagawa, "Monte Carlo filter and smoother for non-gaussian nonlinear state space models", J. Comput. Graph. Statist., 1996.
- [7] M. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filters," J. Amer. Statist. Assoc., vol. 94, no. 446, pp. 590–599, 1999.
- [8] A. Doucet, S.J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering". Stat. & Comp., vol. 10, no. 3, pp. 197-208, 2000.
- [9] P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez, "Particle filtering," IEEE Signal Processing Magazine, vol. 5, pp. 19-38, Sep. 2003.

- [10] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P-J Nordlund, "Particle Filters for positioning, navigation and tracking," *IEEE Transactions on Signal Processing*, Feb 2002.
- [11] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*, Wiley Series in Probability and Mathematical Statistics, 1987.
- [12] J. Schafer, *Analysis of Incomplete Multivariate Data*, Chapman & Hall, 1997.
- [13] A. Kong, J.S. Liu, and W.H. Wong, "Sequential imputation and Bayesian missing data problems". *J. American Statistical Association*, pp. 278-288, 1994.
- [14] M. B. Ghalia and A. T. Alouani, "Observability requirements for passive target tracking", *Proceedings SSST '93.*, pp. 253-257, 1993.
- [15] V.J. Aidala and S.E. Hammel, "Utilization of modified polar coordinates for bearing-only tracking", *IEEE Trans. Automatic Control*, vol. 28, no. 3, pp. 283-294, March 1983.
- [16] N. Peach, "Bearings-only tracking using a set of range parameterised extended Kalman filters," *Proc. Inst. Elect. Eng. Contr. Theory Appl.*, vol. 142, no. 1, Jan. 1995.
- [17] R. Karlsson and F. Gustafsson, "Recursive Bayesian estimation: bearing-only applications", *IEE Proc. Radar Sonar Navig.*, vol. 152, No. 5, October 2005.
- [18] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering for practitioners," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736-746, 2002.
- [19] K. L. Chung, *A Course in Probability Theory: Second Edition*, Academic Press, (1974).
- [20] W. Rudin, *Principles of Mathematical Analysis*, 3rd Edition, McGraw-Hill, 1976
- [21] K. Yosida. *Functional analysis*. Springer-Verlag, Berlin, 6th edition, 1980.

- [22] T. B. Schon, Estimation of Nonlinear Dynamic Systems Theory and Applications. PhD thesis. Linkopings University, 2006.
- [23] P. Del Moral, Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications, Springer, 2004.
- [24] S.-h. Cui and C.-q. Zhu, "Application of Kalman Filter to bearing-only target tracking system", 3rd International Conference on Signal Processing (ICSP'06), vol. 2, pp. 1679-1682, 1996.
- [25] X.R. Li and V.P. Jilkov, "A survey of maneuvering target tracking-part III: measurement models", In proceedings of SPIE Conference on Signal and Data Processing of Small Targets, pp. 423-446, San Diego, USA, 2001.
- [26] J.P. Le Cadre, "Properties of estimability criteria for target motion analysis", IEE Proceedings - Radar, Sonar and Navigation, no. 2, vol. 145, pp. 92 - 99, 1998.
- [27] L. G. Weiss, S.C.A Thomopoulos, "Target detection and localization from bearing-only and bearing/range measurements", Proc. Aerospace Control Systems, May 25-27, 1993, pp. 549-553
- [28] D. Koks, "Passive geolocation for multiple receivers with no initial state estimate", Electronic Warfare Division: Electronics and Surveillance Research Laboratory, DSTO.

④ BC-25-48